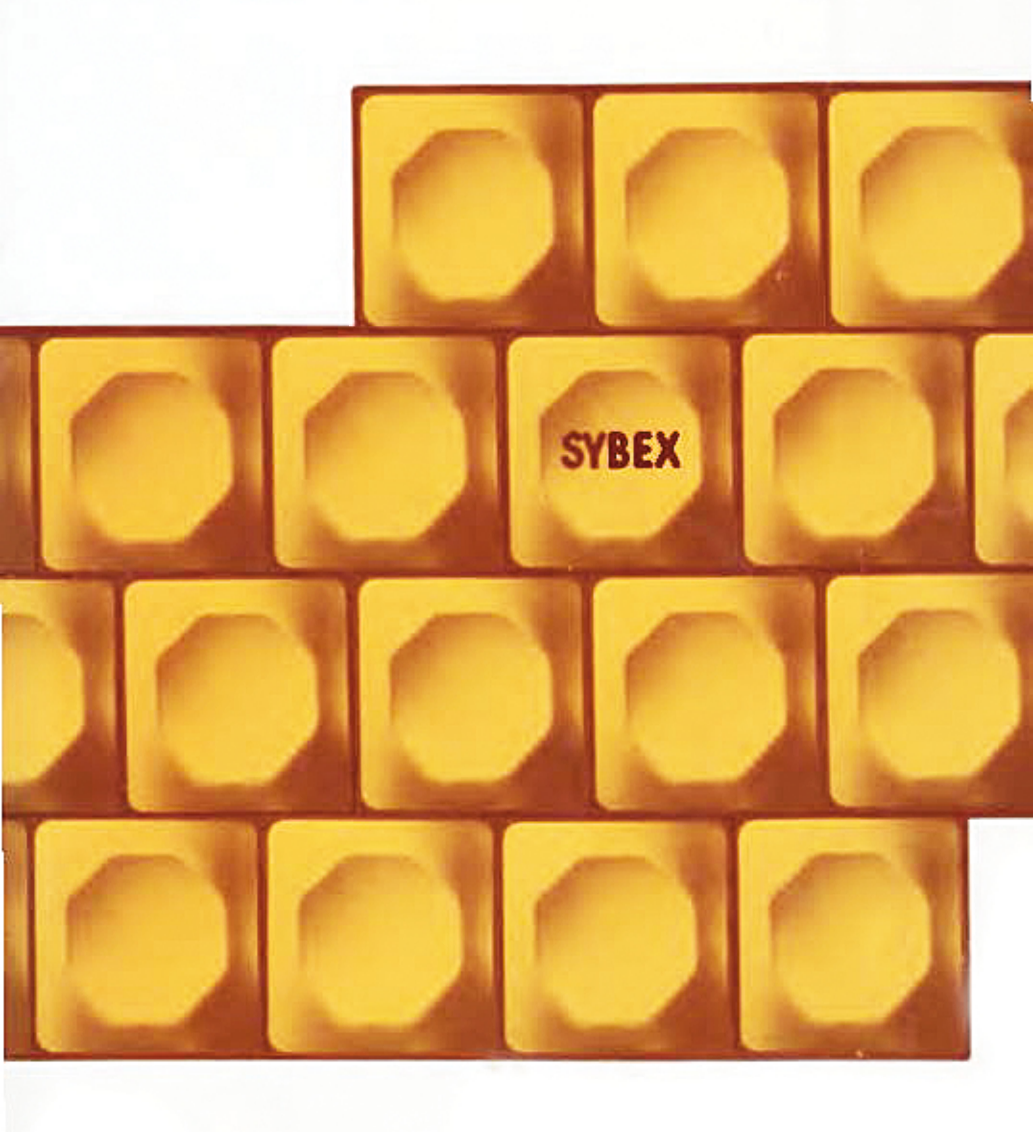
# INTRODUCTION AUX MICROORDINATEURS INDIVIDUELS ET PROFESSIONNELS

RODNAY ZAKS



# INTRODUCTION AUX MICROORDINATEURS INDIVIDUELS ET PROFESSIONNELS

RODNAY ZAKS




U.S.A.
2020 Milvia Street
Berkeley, California 94704
Tel. : (415) 848-8233
Telex : 336311

EUROPE
313 rue Lecourbe
75015 - Paris, France
Tél. : (1) 828-2502
Telex : 200858

Traduction française:

Daniel Jean David, Pierre Le Beux

Illustrations:

Daniel Lenoury



ISBN: 2-902414-12-9

# TABLE DES MATIERES

**PREFACE** .......................................... **7**

**I. L'ERE DES MICROORDINATEURS** .......... **8**

*A la maison. Au bureau. A l'hôpital. De retour à la maison. La cité électronique. Quand ? Les microordinateurs.*

**II. UTILISONS LE SYSTEME** .................... **18**

*Introduction. Connectons les éléments. Jouons un peu. Un autre programme. Récapitulation.*

**III. DEFINITIONS FONDAMENTALES** .......... **27**

*Applications des microordinateurs. Définitions de base. La fabrication d'un microprocesseur. L'UC. Les bus. La mémoire. Les entrées-sorties. Un système microprocesseur. Le système microordinateur. Microordinateur ou miniordinateur. Avantages des microordinateurs. Récapitulation.*

**IV. COMMENT IL FONCTIONNE** ............... **47**

*Introduction. L'architecture de base d'un système : l'UC, la mémoire, les entrées-sorties. L'alimentation. Récapitulation.*

**V. LA PROGRAMMATION** .................... **61**

*Définitions de base. Langages de programmation. Organigrammes. Représentation de l'information. Opérations logiques. Représentation externe. Développons un programme. Logiciel nécessaire pour le développement d'un programme.*

**BASIC ET APL** .............................. **76**

*Lisons des valeurs. Commentaires. Notation scientifique. Conditions de test. Boucles automatiques. Fonctions fournies. Sous-programmes. Fonctions définies par l'utilisateur. Listes et tableaux. Exemples. Une tabulation commerciale. Facilités spéciales. Votre Basic est-il complet ? Le Basic de gestion. Quelle est la vitesse de votre Basic ? Le Basic compilé. Les limitations de Basic. Alternatives à Basic-APL. Exemples en APL. Le langage naturel. Sommaire des instructions Basic.*

**APPLICATIONS DE GESTION** .............. **96**

*Introduction. Applications des ordinateurs en gestion. Traitement de texte. Utilisation d'un système informatique commercial. Les impératifs d'un système de gestion : comptes créditeurs, comptes débiteurs, inventaire, mise à jour, fichier d'adresses. Facilités logicielles existantes. Utilisation des microordinateurs en gestion. Et les miniordinateurs ? Programmes commerciaux sur mesures. Récapitulation. Achat d'un système : sommaire. Facilités souhaitables en gestion. Une liste de vérification.*

**CHOIX D'UN SYSTEME** ..................... **121**

*Introduction. Critères de sélection. Evolution des prix. Performance. Benchmarks. Les critères importants. Matériel complet. Logiciel complet. Facilité d'emploi. Performance d'ensemble du système. Fiabilité. Types de systèmes : kit, carte ou système ? Microordinateur sur une carte, système microordinateur. Panneau avant ou pas ? Quel bus ? En kit ou tout assemblé ? Les connecteurs à l'arrière du coffret. Récapitulation. Et le microprocesseur XYZ ?*

**LES PERIPHERIQUES** ....................... **141**

*Le clavier. L'affichage CRT. Le moniteur vidéo. Le terminal à écran vidéo. Le poste télé traditionnel. Représentation de texte sur un écran. Combien de lignes et de caractères ? Autres caractéristiques des écrans. Stupide ou intelligent. Récapitulation des écrans. Communication avec l'écran. L'imprimante à bande. La Selectric. L'imprimante à marguerite. L'imprimante à aiguilles. L'imprimante-lignes. Super-imprimantes. Imprimantes de gestion. Les disques. Le disque souple, le mini-disque souple. Augmentation de capacité. Un ou deux disques ? Bandes magnétiques. Mémoires de masse futures. Le crayon lumineux. Le manche à balai. Tablette digitaliseur. Entrée vocale. Sortie vocale. Diodes LED. Interrupteurs. Relais. DAC et ADC. Récapitulation.*

X. CHOIX D'UN MICROORDINATEUR .......... 171

*Microhistorique des ordinateurs. La quatrième génération. Microordinateurs commerciaux. Microordinateurs en une seule carte. Le Kim I, le NEC TK80. Systèmes inégrés: Videobrain, Bally, Pet, TRS80. Les microordinateurs à usage personnel: Apple II, Altair, Imsai, Sol, Cromenco, etc. Petits systèmes de gestion.*

XI. LES COUTS D'UN SYSTEME DE GESTION ... 207

*Le coût réel d'un système. Les coûts cachés. Quand acheter. Le temps partagé. L'achat d'un système : une récapitulation.*

XII. COMMENT ECHOUER AVEC UN SYSTEME DE GESTION .......... 212

*Introduction. Pannes du matériel. Techniques d'amélioration de fiabilité. Pannes du logiciel. Procédures : précision des données, contrôle des procédures, digits de contrôle, sécurité des données, le choc-ordinateur. Récapitulation.*

XIII. AU SECOURS .............................. 221

*Sources d'information. Publications, clubs, boutiques ordinateurs. Formation. Expositions microordinateurs.*

XIV. DEMAIN .................................. 227

*Récapitulation. Demain : prix et miniaturisation. Conclusion.*

APPENDICES :
A. Logique des ordinateurs .......................... 229
B. Bits et octets .................................. 235
C. Principes des communications ..................... 239
D. Fichiers ........................................ 242
E. Fabricants de systèmes de gestion ................. 247
F. Fabricants de microordinateurs .................... 248

# PREFACE

Ce livre a été écrit pour le lecteur qui ne connaît encore rien sur les ordinateurs. Il ne nécessite pas de connaissances électroniques ou techniques.

Il s'agit d'un livre pratique et éducatif introduisant progressivement tous les éléments constituant un système informatique. Il a été conçu pour une lecture facile, et le lecteur intéressé par la technique trouvera des informations supplémentaires dans les appendices en fin d'ouvrage. Les trois premiers chapitres vous introduisent au monde des microordinateurs. Vous parlerez « ROM » et « RAM » et comprendrez les fonctions d'un système réel.

Pour celui qui veut comprendre comment cela fonctionne, le chapitre IV présente le mécanisme interne de fonctionnement. Cependant cela n'est pas indispensable.

Si vous voulez programmer, vous aurez le choix entre deux niveaux de langages : le langage assembleur et le langage évolué. Les chapitres V et VI vous introduiront à ces techniques.

Le chapitre VII présentera les spécifications d'un système professionnel ou commercial, un champ nouveau d'application des systèmes microordinateurs.

A ce stade vous aurez acquis les connaissances de base nécessaires à l'analyse des choix possibles.

Un des choix les plus importants du point de vue du coût n'est sans doute pas celui du microordinateur proprement dit mais celui des périphériques. Le chapitre IX vous présentera tous les types de périphériques qui peuvent être utilisés et donnera les raisons de l'utilisation de tel ou tel périphérique dans des circonstances spécifiques.

Le chapitre X présente les caractéristiques des principaux microordinateurs actuellement sur le marché.

Connaissant le matériel disponible, la question est alors de savoir s'il faut acheter un système. Le chapitre XI explorera les alternatives disponibles pour un utilisateur commercial.

Les pièges dans lesquels peuvent tomber un utilisateur sont nombreux et le chapitre XII décrit les principales causes des échecs possibles.

Le chapitre XIII fournit des renseignements d'ordre général pour aider le lecteur à se documenter et s'informer.

Enfin, une prévision de l'évolution de ce nouveau marché est présentée au chapitre XIV.

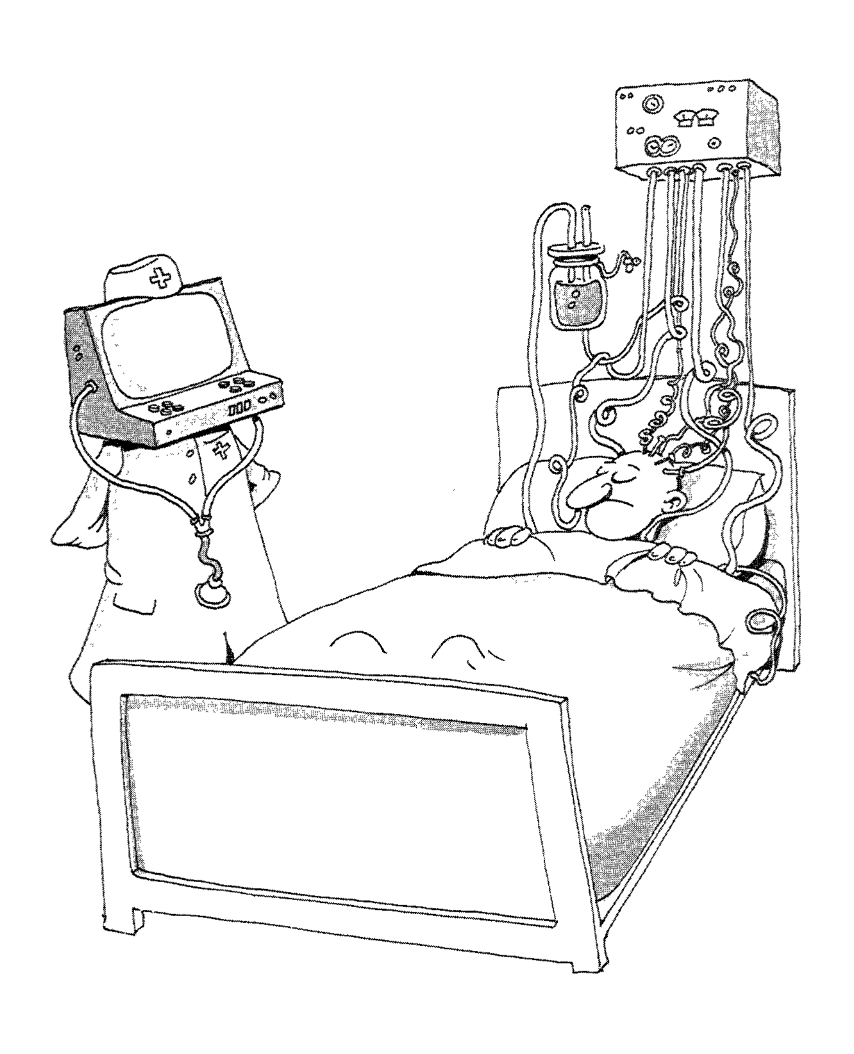
# CHAPITRE I
# L'ERE
# DES MICROORDINATEURS

**A la maison, quelque-part aux Etats-Unis**

Il est 7 heures du matin. Le réveil joue un air de musique douce dans la pièce. Jim se réveille et arrête le réveil. Un écran de télévision miniature s'allume près du lit, et affiche un message : « consulter le fichier RV avant votre départ ». Jim se lève. Avant de quitter la chambre à coucher, il appuie en silence sur deux touches du clavier à son chevet. « PR ». Il s'agit de l'ordre de mise en route du petit déjeuner. La machine à faire le café se met en marche automatiquement à la cuisine, tandis que les deux croissants sont réchauffés. La confirmation apparaît sur le petit écran près du lit : « CAFE EN ROUTE, JE RECHAUFFE DEUX CROISSANTS ».

Jim se rend maintenant dans la pièce qui sert d'étude, afin de consulter le fichier, ainsi que l'ordinateur le lui a demandé. Au fur et à mesure qu'il traverse la maison, les lumières s'allument l'une après l'autre sur son passage.

Jim a maintenant atteint la pièce où se trouve le terminal. Le contenu du fichier RV est déjà affiché pour lui sur le grand terminal à écran TV situé sur le bureau. Il s'agit de deux nouveaux rendez-vous arrangés pour lui au bureau après qu'il soit parti : il découvre qu'un « message prioritaire » a été envoyé par son chef de service durant la soirée, lui demandant de se rendre d'urgence à une réunion exceptionnelle de ventes ce matin même à 9 heures. Jim prend note de son nouvel horaire de rendez-vous et se met à interroger nerveusement son microordinateur familial afin de découvrir qui d'autre va participer à cette réunion. Malheureusement, la liste des participants à la réunion, qui a été définie par son chef de service, a été protégée contre une lecture non autorisée, et ne peut être affichée sur l'écran. Jim se lève de son siège et décide de simplement se dépêcher et d'arriver au bureau en avance.



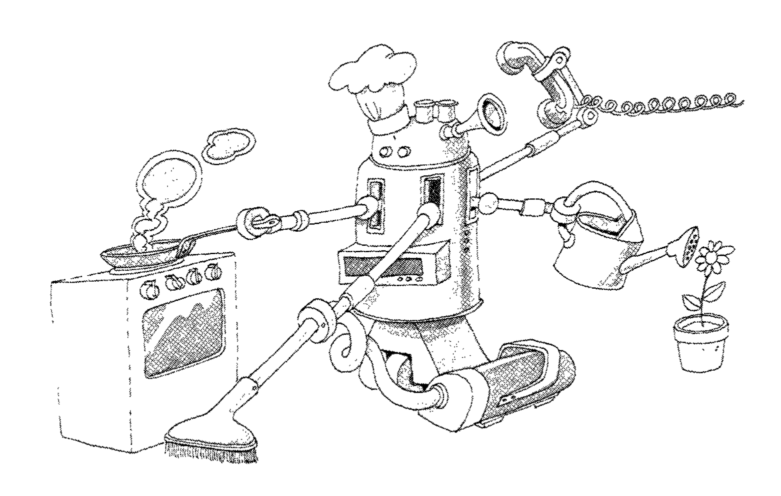
Son déjeuner est déjà prêt et l'attend à la cuisine. Le chauffage pour la maison s'est allumé automatiquement et la salle à manger est agréablement chaude.

Jim déjeune rapidement, d'une manière nerveuse, tandis que sa femme et ses enfants continuent de dormir. Dès qu'il a terminé il appuie sur les touches du clavier de la cuisine pour sa voiture. La porte du garage s'ouvre et le moteur de sa voiture se met en marche automatiquement.

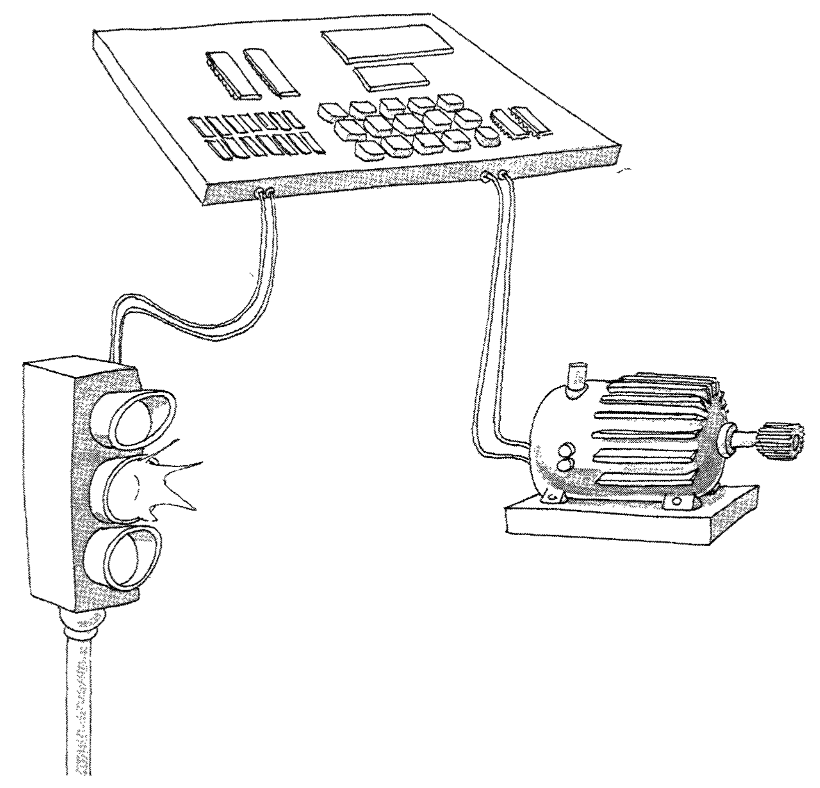
Jim va vite encore une fois au terminal principal dans son étude et interroge rapidement les microordinateurs domestiques de deux de ses collègues. Il découvre qu'ils sont déjà arrivés au bureau. Il se dépêche alors de partir. Il se rend au garage, où le moteur de sa voiture est maintenant chaud, et démarre rapidement. Les lumières dans la maison s'éteignent automatiquement lorsqu'il est parti.

Une demi-heure plus tard, une séquence analogue se déroulera quand sa femme se réveillera.

Retrouvons Jim, dans sa voiture. Il voudrait se rendre à son bureau aussi vite que possible. Il utilise le clavier de sa voiture pour interroger l'ordinateur urbain de contrôle de circulation. Il est alors informé d'un détour recommandé par une voie parallèle, afin de gagner du temps sur l'itinéraire recommandé jusqu'à son bureau. Malheureusement, Jim n'est pas familier avec l'itinéraire de détour recommandé. Il demande de l'aide. L'ordinateur de sa voiture va maintenant interroger les détecteurs de carrefour et va automatiquement afficher quand il faut tourner à gauche ou à droite pour rester



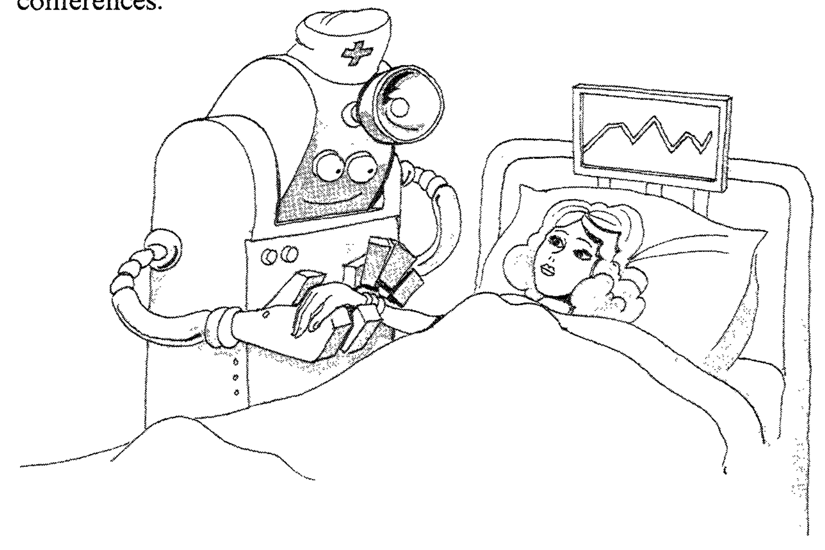
sur l'itinéraire conseillé. Ainsi, l'affichage de sa voiture clignote en ce moment : « tourner à droite prochaine intersection ». Jim suit les instructions reçues. Il inspecte rapidement les indications de l'ordinateur quant au trajet à parcourir : encore 12 km avec un temps moyen estimé de 20 minutes. Jim se sent mieux à présent : il arrivera au bureau largement à temps pour la réunion. Il n'a pas à se soucier de problèmes mécaniques prévisibles sur sa voiture : l'ordinateur de sa voiture a déjà accompli un test complet du moteur pendant que celui-ci chauffait, et a vérifié que tous les organes moteurs étaient en bonne condition. Le calculateur kilométrique a également vérifié que la quantité d'essence contenue dans le réservoir était suffisante pour la durée habituelle du trajet.



**Au bureau**

Jim est maintenant arrivé à son bureau. Il se dirige vers la porte, et place la paume de sa main sur un emplacement rectangulaire

réservé à cet effet, qui est équipé d'un capteur spécial. La porte s'ouvre. L'identité de Jim a été vérifiée par l'ordinateur de sécurité, et ce dernier enregistre en même temps son entrée dans les locaux. Sur son bureau, Jim trouve une liste imprimée par l'ordinateur qui donne l'ordre du jour de la réunion de ventes de 9 heures. Ainsi averti du but de la réunion, il se rend aussitôt à son terminal individuel, et demande l'impression immédiate de tous les fichiers de données dont il voudrait pouvoir disposer en cours de réunion. Il constate également que la liste des participants a maintenant été débloquée : Jim peut l'afficher sur son écran et vérifier qui d'autre va participer. Une mise à jour illumine brusquement l'écran : la réunion est repoussée d'un quart d'heure. Jim décide d'appeler Pierre, un ami et collègue en qui il a confiance, pour discuter avec lui de leurs résultats de ventes à ce stade. Il appuie sur la touche de son téléphone marquée « Pierre ». Au même moment, un petit écran s'illumine sur son téléphone, et un message apparaît, indiquant l'arrivée d'un appel : « M. Gorvin de la société Superprécis appelle ». Il appuie sur la touche « refuse ». Un message va être envoyé en réponse à l'appel de M. Gorvin, lui indiquant que Jim est en ligne, et lui offrant la possibilité de laisser un message enregistré ou de s'adresser à un collègue de Jim. L'affichage silencieux de l'appel de M. Gorvin a permis à Jim d'effectuer une décision immédiate sans interrompre sa conversation avec son collègue Pierre. Jim finit sa conversation et se rend dans la salle de conférences.



**A l'Hôpital**

Pendant ce temps, sa femme Linda, qui travaille maintenant surtout à la maison comme architecte, décide de rendre visite ce matin à son amie Janette, avant de se mettre au travail. Elle prend sa voiture jusqu'à l'hôpital. A l'hôpital, l'hôtesse tape son nom sur le terminal d'accueil, qui indique que la visite est acceptée, à la fois en fonction du règlement de l'hôpital et des consignes du médecin quant aux visites. Le numéro de la chambre où se trouve son amie apparaît sur l'écran. C'est la pièce 305 au troisième étage. Elle s'y rend donc. Janette vient de subir une opération sérieuse et est maintenant sous surveillance par ordinateur. Des sondes spéciales surveillent en permanence ses signes vitaux, tels que son rythme cardiaque, la pression sanguine, l'activité cérébrale, le rythme respiratoire, sa température, et d'autres fonctions encore.

Ces sondes sont connectées à un microordinateur de chevet qui compare continuellement les valeurs mesurées aux limites normales telles que le médecin traitant les a définies, et vérifie aussi les combinaisons de valeurs qui pourraient indiquer un problème physiologique. Jusqu'à présent, toutes les indications sont normales. Toutefois, dans la pièce à côté, le programme de surveillance cardiaque a détecté cette nuit une arythmie, indiquant de manière probable l'imminence d'une crise cardiaque. Le signal d'alarme cardiaque immédiate a réveillé Janette en sursaut. Elle a pu entendre les médecins se précipiter dans la pièce et entreprendre immédiatement une action médicale préventive pour éviter que la crise cardiaque possible ne se déclenche. L'intervention a complètement réussi, et l'attaque a été prévenue.

Janette explique à Linda que lorsqu'elle quittera l'hôpital, sa condition cardiaque devra aussi être surveillée pendant au moins deux mois. Suivant les conseils de son médecin, elle a décidé de louer une unité portable à l'hôpital. Cet appareil déclenchera un signal à l'avance si une crise éventuelle était prévue par le microordinateur portable. De plus, chaque soir, elle devra brancher son microordinateur portable au téléphone, afin qu'il puisse transmettre à l'ordinateur de l'hôpital les données recueillies et accumulées pendant la journée. Ce dernier exécutera alors un programme complexe de diagnostic, afin d'établir soit un progrès normal de convalescence, ou bien les signes anticipés de complications possibles.

L'ensemble des médicaments et traitements de Janette ne sont pas

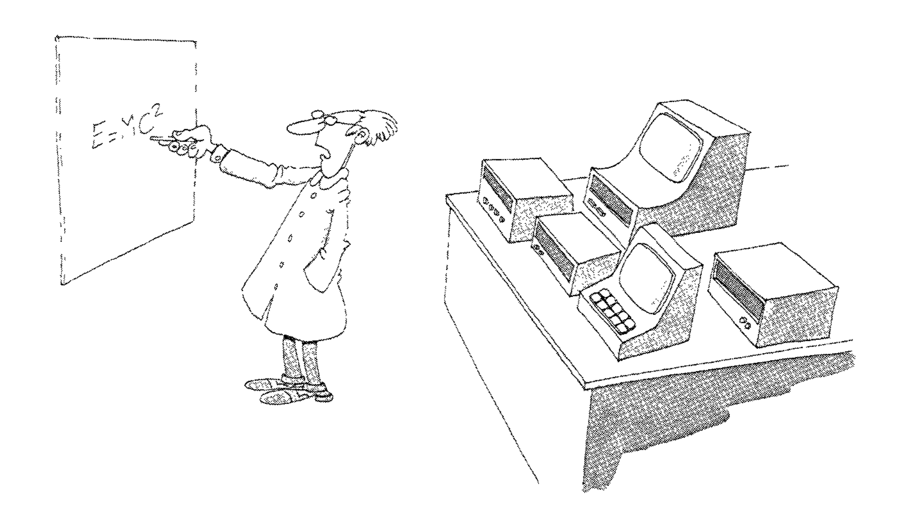
surveillés par son ordinateur de chevet, mais par un puissant ordinateur d'hôpital qui vérifie la prescription de chaque médecin en fonction des médicaments que Janette prend déjà, de la liste de leurs incompatibilités et de la liste des produits auxquels Janette pourraît être allergique. Toute combinaison qui pourraît être dangereuse est immédiatement portée à l'attention du médecin traitant.

Janette qui est normalement très active, a emmené son microordinateur personnel dans la pièce, afin de continuer lentement à travailler sur le rapport qu'elle prépare depuis plusieurs semaines. Elle peut utiliser son microordinateur pour communiquer avec les ordinateurs bibliothécaires, lui permettant ainsi d'examiner sélectivement le contenu de livres qu'elle souhaite consulter. Elle peut ainsi poursuivre son travail de préparation du rapport à sa propre allure.

Elle fait attention de ne pas utiliser son microordinateur trop longtemps, car son médecin lui a interdit de travailler plus de trois heures : en cas de dépassement, son microordinateur personnel est automatiquement débranché par un ordinateur de surveillance.

**De retour à la maison**

Linda reprend sa voiture et rentre chez elle, où elle a l'intention de se débarrasser rapidement de quelques tâches ménagères, puis de se remettre à son travail d'architecture. Une fois de retour, elle consulte son microordinateur personnel pour vérifier la liste des choses urgentes à faire aujourd'hui. Elle s'aperçoit qu'elle doit transférer de l'argent de son compte d'épargne, car son compte chèques est pratiquement à zéro. De plus, plusieurs factures doivent être acquittées immédiatement. Elle décide donc de s'occuper d'abord de cela. Elle se branche sur l'ordinateur de la banque, et donne ses instructions pour que les transferts appropriés soient effectués de son compte à celui des créditeurs. Pour chaque transaction, elle tape un numéro de code qui vérifie que la personne ordonnant la transaction a l'autorisation de le faire. Ensuite, elle se branche sur l'ordinateur de son magasin favori afin d'ordonner la livraison de toute une liste d'objets et denrées. Toutefois, ce matin elle n'arrive pas à décider quels fruits et légumes acheter. Elle demande une visualisation directe des denrées disponibles, et l'affichage des prix. Sur son écran de télévision, les fruits et légumes disponibles apparaissent maintenant, ainsi que leur prix. Elle prend quelques notes, puis passe le reste de ses commandes. Le tout sera livré avant




4 heures cet après-midi, ainsi qu'il est indiqué dans un rectangle en haut à droite de son écran. Linda a maintenant pris soin des besoins les plus pressants de la journée, et décide de consacrer une heure à sa leçon d'Espagnol. Les phrases en Espagnol résonnent dans la pièce. Les sons sont engendrés par programme, grâce à un synthétiseur vocal. En même temps, le texte écrit correspondant et les images apparaissent sur l'écran. Elle doit répéter les mots et les phrases. Chaque fois qu'elle n'arrive pas à répéter correctement un son, le programme reprend le mot et l'oblige à le répéter jusqu'à ce qu'elle l'imite convenablement, dans la limite de cinq essais. Le programme continue en séquence, avec des exercices de difficulté graduée. Au bout d'une heure, Linda décide de s'arrêter pour aujourd'hui, de déjeuner, puis de se mettre au travail sur ses dessins d'architecture. Avant de quitter le terminal, afin de déjeuner, elle spécifie une liste d'évènements qui pourront déclencher automatiquement le système d'appel à la cuisine. Elle autorise les appels de son mari, de Janette à l'hôpital, de ses enfants, de son patron, et les dernières nouvelles du résultat des élections à la mairie. Puis elle se rend à la cuisine.

Plus tard, elle travaillera à partir de sa chambre, dessinant ses plans architecturaux sur une tablette spéciale, les voyant apparaître sur l'écran, puis les corrigeant à l'aide de son ordinateur. Quand les plans seront en forme définitive, elle les transmettra à l'ordinateur de son bureau, où ils vont être examinés, et critiqués ou approuvés par

son patron. Plus tard dans l'après-midi, après avoir terminé son travail, l'écran s'illumine, et deux messages y apparaissent : deux de ses amies ont appelé pendant l'après-midi, mais ont été empêchées d'interrompre son travail. Elles ont dû laisser un message, qui a été mis en mémoire, et qui apparait maintenant. Les enfants vont bientôt revenir de l'école, et vont aussi utiliser l'ordinateur de Linda pour préparer leurs devoirs. Linda compte acheter bientôt à sa fille aînée son propre ordinateur.

**La cité électronique**

Tous les microordinateurs dans la maison de Jim, ainsi que dans toutes les autres maisons, les bureaux, les hôpitaux, et les autres bâtiments, sont interconnectés en réseaux complexes. L'information est désormais constamment disponible pour tous ceux qui ont le droit d'en disposer, et ceci instantanément, à tout moment. Les transferts d'information sont pratiquement instantanés, et peuvent être accomplis à partir de n'importe quel terminal, à la maison, dans la voiture, ou au bureau. Les habitudes de travail ont été changées d'une manière radicale. Beaucoup moins de gens travaillent au bureau, à moins qu'ils n'aient besoin de communiquer personnellement avec l'un de leurs collègues ou d'utiliser des ressources spéciales disponibles uniquement au bureau. La plupart du temps le travail est accompli à la maison ou en d'autres endroits, du moins pour toutes les tâches qui peuvent être accomplies à l'aide d'un terminal. Toutes les procédures qui pouvaient être automatisées dans la maison ou la voiture, ainsi que les systèmes de communication tels que le téléphone, ou toutes les autres séquences prévisibles sont maintenant accomplies et gérées par microordinateur miniaturisé. Ainsi, l'ancienne montre bracelet est maintenant devenue un petit microordinateur équipé d'un clavier miniature et d'un affichage qui peut être utilisé pour communiquer avec n'importe lequel des autres microordinateurs. Le travail est devenu plus efficace et plus créatif, résultant en une semaine de travail de trois ou quatre jours.

**Quand ?**

Le scénario ci-dessus est fictif. Quand sera-t-il une réalité ? *Toutes les possibilités que nous venons de décrire peuvent techniquement être réalisées aujourd'hui !*

Simplement, pour des raisons économiques, elles ne sont pas encore réalisées à grande échelle, où déjà disponibles à un coût suffisamment bas. Toutefois, il est possible de prédire en toute sûreté que la plupart d'entre elles seront devenues une réalité dans les dix ans qui viennent. Bien sûr, beaucoup d'autres encore apparaîtront, auxquelles nous ne pensons même pas encore. Le but de cette introduction était de donner une idée de la variété des applications aujurd hui rendues possible grâce aux progrès des microordinateurs.

**Les microordinateurs**

Les microordinateurs résultent du progrès surprenant de la technologie MOS (Metal Oxyde Semiconducteur) qui permet désormais de réaliser un ordinateur complet sur un petit rectangle de silicium (une « pastille » d'environ 5 mm de côté). Les fonctions qui étaient jadis accomplies par de gros ordinateurs remplissant toute une pièce peuvent désormais être accomplies par les microordinateurs en un boîtier. D'autres applications techniques d'utilisation seront présentées au long de ce livre.

Apprenons maintenant à utiliser le microordinateur de Jim ou de Linda, dès aujourd'hui !



# CHAPITRE II
# UTILISATION DU SYSTEME

## INTRODUCTION

Nous venons juste d'acquérir un système microordinateur. Il comprend : une boîte contenant le microordinateur, une unité de visualisation de type télévision associée à un clavier, et un enregistreur à cassettes ordinaire. Bien sûr, il existe de nombreuses autres options et elles seront présentées dans les chapitres suivants. Le but de ce chapitre est de comprendre ce que sont les éléments principaux du système.

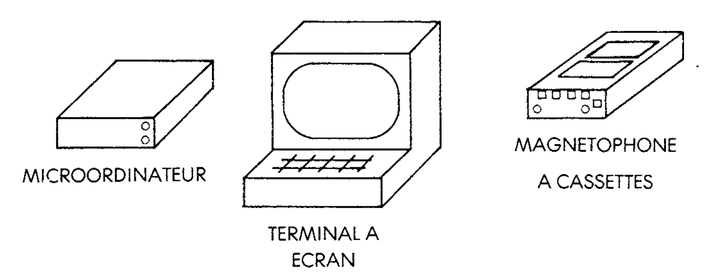
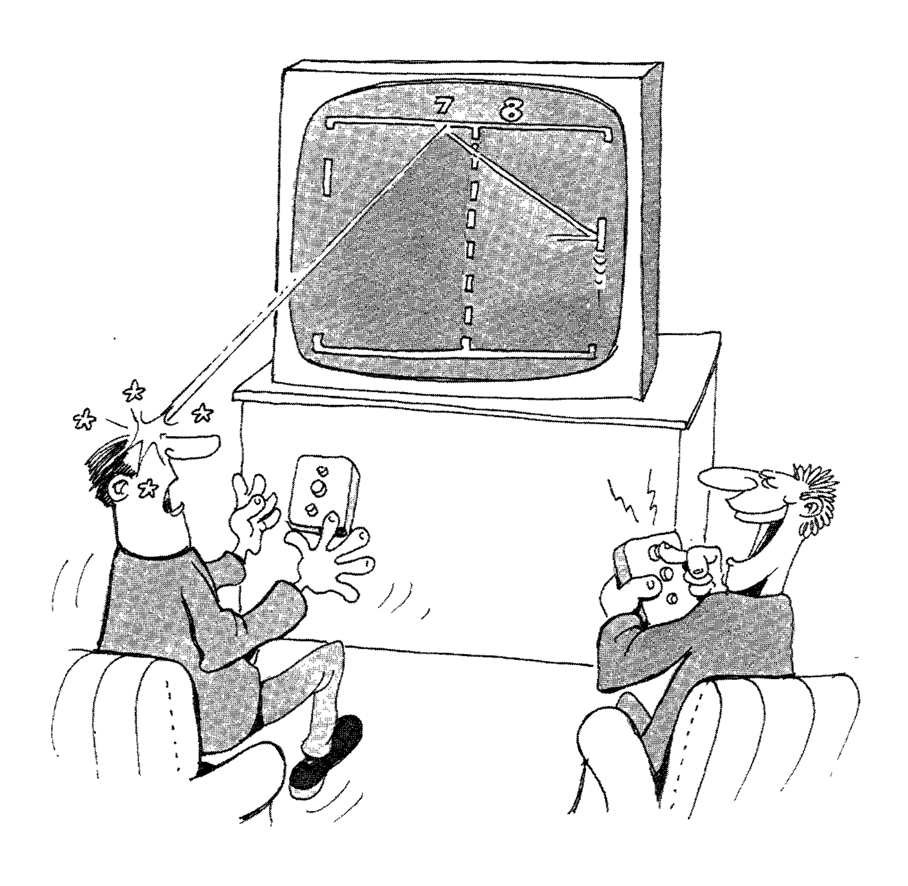



Fig. 2.1 : Un système microordinateur personnel

## LA CONNEXION DES ELEMENTS

Pour satisfaire notre impatience à faire quelque chose avec le système, il faut d'abord connecter les différents éléments. Nous disposons de trois modules et de trois câbles. La boîte microordinateur exécute les programmes, l'unité de télévision avec son clavier est appelée « terminal de visualisation à tube cathodique » (CRT = cathode ray tube). Le mot « terminal » indique qu'il inclut un clavier avec des fonctions spéciales (décrites plus tard).

L'enregistreur de cassettes est un magnétophone à cassettes ordinaire. Il dispose d'un compteur et d'une commande à distance.


7 8

Connectons d'abord le terminal de visualisation au microordinateur. Le câble d'interconnection est terminé par un connecteur standard « RS 232 » qui assure l'interface entre le microordinateur et l'unité de visualisation. Presque tous les microordinateurs utilisent cet interface pour la connection à des unités de visualisation. Cependant certains systèmes intégrant l'unité de visualisation à la boîte microordinateur peuvent s'en passer (le « PET » de Commodore en est un exemple).

La connection de l'enregistreur nécessite un câble spécialisé.

Il est connecté à la prise d'entrée du microphone, la sortie du haut-parleur et la prise de commande à distance du côté du magnétophone, et à une prise spéciale du côté microordinateur.




Fig. 2.2 : Ce système intègre le clavier à la boîte microordinateur

Maintenant mettons le tout en marche :

— Le microordinateur a simplement un bouton de mise sous tension que l'on enfonce, ce qui allume une lampe diode émettrice de lumière (LED) : « En marche ».

— Le terminal de visualisation est mis sous tension et un carré blanc apparaît sur l'écran.

Ce carré est un « curseur » qui indique la position du prochain caractère à apparaître sur l'écran. En appuyant sur le bouton « remise à zéro » (« reset »), le curseur se positionne en haut à gauche de l'écran. Ceci est l'endroit où nous (ou le microordinateur) commencerons à écrire sur l'écran.

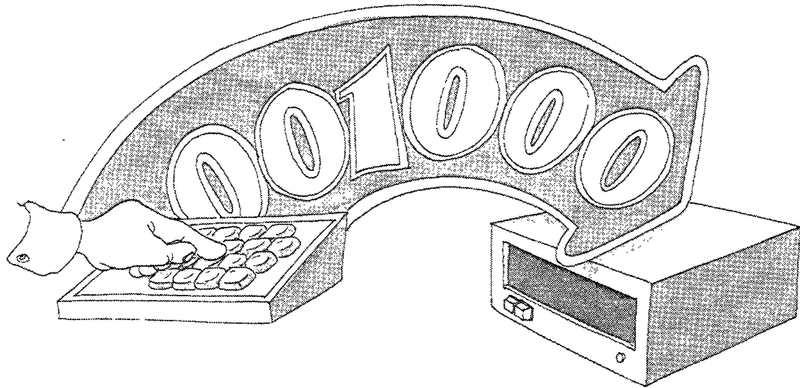



Fig. 2.3 : L'action d'appuyer sur une touche du clavier envoie un code binaire vers le microordinateur

On peut maintenant appuyer sur n'importe quelle touche du clavier. Le calculateur répond immédiatement : le mot « PRET » (« READY ») apparaît sur l'écran.

Le système est maintenant prêt à accepter des commandes et se met en attente indéfinie. Quand on appuie sur une touche, l'électronique du clavier détecte un contact et associe un code binaire à la touche. Le microprocesseur reçoit un signal, lui demandant de lire un caractère. Il le fait par l'intermédiaire du câble d'interface RS 232. Dans le cas du démarrage, l'action effectuée par l'ordinateur ne dépendait pas du caractère frappé : il visualise dans tous les cas le mot « PRET ».

Le mot « PRET » est transmis à partir de la mémoire du microordinateur vers l'unité de visualisation sous forme de quatre caractères consécutifs : P-R-E-T.

Sur l'écran on peut maintenant observer : la lettre « A » qui a été frappée et renvoyée en écho par le microordinateur. Sur la ligne suivante, on lit le mot « PRET » à nouveau.

En fait le microordinateur a du envoyer deux caractères supplémentaires de façon à ce que le mot « PRET » n'apparaisse pas sur la même ligne que « A ». Le premier caractère est appelé « avance ligne » (« line feed ») et fait déplacer le curseur d'une ligne vers le bas.

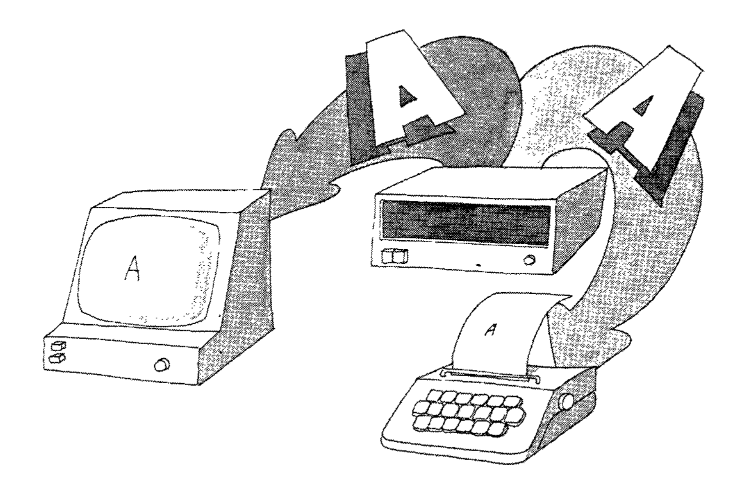



Fig. 2.4 : La réponse du microordinateur

Le deuxième caractère est le « retour chariot » (« carriage return ») qui positionne le curseur sur la position la plus à gauche de la ligne. Ensuite les caractères du mot « PRET » sont envoyés.

Le mécanisme de l'écho est une caractéristique importante des systèmes de dialogue avec un ordinateur. En effet sur une machine à écrire traditionnelle, une lettre est frappée dès que l'on a appuyé sur une touche. Sur un terminal connecté à un ordinateur, que ce soit un téléscripteur ou une unité de visualisation, un caractère est normalement imprimé ou visualisé *par l'ordinateur.*

Ceci a deux avantages :

1 - Vérifier que la transmission s'est effectuée sans erreur. La visualisation de « A » sur l'écran garantit que le caractère frappé a bien été lu par l'ordinateur et que le câble de connexion fonctionne bien dans les deux sens.

2 - Le jeu de caractères peut être *interprêté.* Ainsi dans un système commercial, un « mot de passe » peut être demandé avant que l'utilisateur puisse accéder à ce système. Dans ce cas il serait regrettable que le mot de passe soit visualisé lorsque l'on appuie sur les touches ! Il faut donc ne pas renvoyer l'écho des caractères correspondant à un mot de passe (sécurité). Tout bon système doit garantir cela.

Nous allons quitter pour le moment l'unité de visualisation pour jouer à un jeu.

## LANCEMENT D'UN JEU

Nous allons jouer au jeu de « morpion » avec le calculateur. Il faut d'abord charger un programme dans la mémoire du microordinateur à partir de la cassette où se trouve le programme de jeux.

Cette tâche est rendue nécessaire par le fait que la mémoire principale du microordinateur est volatile. Après la mise sous tension, elle ne contient rien encore.

Les programmes sont donc stockés de façon permanente sur des supports magnétiques tels que des cassettes dont le contenu n'est pas affecté par une coupure de tension.

Après avoir allumé le système, les programmes contenus sur la cassette pourront être transférés dans la mémoire centrale pour y être exécutés. L'utilisateur, assis devant le terminal, frappera les commandes nécessaires, définies par le manuel de l'utilisateur, pour transférer le programme « morpion » dans la mémoire de l'ordinateur. Cette séquence de commandes dépend du constructeur qui a défini le système. Une fois le programme chargé, le manuel d'utilisation du programme indique comment l'utiliser et jouer avec l'ordinateur.

Ici, après avoir tapé la commande « CHARGER MORP A PARTIR DE LA BANDE », un damier de neuf carrés apparaît sur l'écran. Nous sommes prêts à jouer. Il faut maintenant indiquer nos coups. L'entrée de ces indications fournit des données au programme qui répond en fonction de ces données. Les données sont entrées ici en spécifiant dans quel carré on veut entrer un « X » (dans ce programme, l'ordinateur joue toujours avec un « O », et nous avec un « X »). Si l'on veut jouer au centre du damier on spécifie « 2-2 », ce qui représente les coordonnées de la ligne 2 et de la colonne 2.

L'entrée est terminée en appuyant sur la touche « retour chariot » du clavier. Ce caractère indique au programme que l'entrée est terminée et qu'il peut continuer.

Ce programme est assez rapide et en moins d'une seconde de « réflexion », il envoie une réponse. Un cercle apparaît dans la position 1-1. C'est à nouveau à notre tour de jouer. Le jeu continue jusqu'à ce que l'ordinateur ou l'utilisateur ait gagné. Ce programme est un programme relativement simple et il peut s'exécuter rapidement de façon à donner des réponses quasi-immédiates.

Dans le cas d'un jeu comme le jeu d'échecs, la réponse de

l'ordinateur pourrait être beaucoup plus lente car il faut exécuter un programme assez long avant d'obtenir une réponse.

Nous venons d'illustrer la fonction d'un « organe d'entrée », ici le clavier, la fonction de l'ordinateur, qui exécute un programme et peut lire des données d'entrée et sortir des réponses. La fonction d' « organe de sortie » est ici accomplie par l'unité de visualisation (écran) qui permet de visualiser les coups choisis par le programme.

Enfin, on a vu qu'une bande magnétique peut être utilisée comme organe de « mémoire à long terme » pour stocker des programmes. Si l'on désirait jouer un autre jeu (le jeu de « NIM » par exemple), il faudrait charger un programme différent en spécifiant des commandes différentes.

A la fin d'un jeu l'ordinateur visualise le message « VOULEZ-VOUS JOUER ENCORE UNE FOIS ». Si nous répondons « NON » l'ordinateur répond : « PRET ». Cela signifie qu'il est revenu au mode « moniteur » (ou « exécutif ») où il attend des commandes venant de l'utilisateur.

Testons maintenant notre connaissance des tables de multiplication et chargeons un autre programme en mémoire.

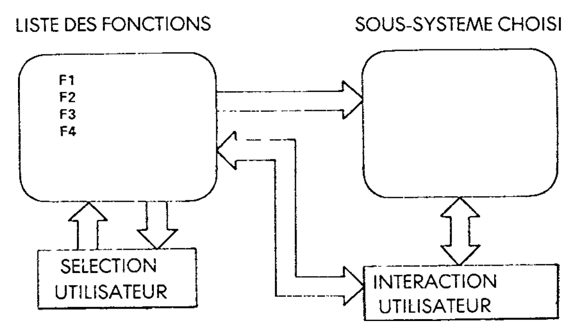



Fig. 2.5 : Le système propose un menu.

## UN PROGRAMME DE CALCUL

Dans le but de charger ce programme, on frappe la commande « CHARGER MATH A PATIR DE LA BANDE », suivie d'un « retour chariot ». Mais cette fois cela ne marche pas et l'ordinateur répond :

« ERREUR - REESSAYER SVP »

En effet, l'ordinateur n'a pas reconnu la commande précédente car il y a une erreur de frappe : « A PATIR » doit être remplacé par « A PARTIR ».

Bien sûr un « ordinateur intelligent » devrait comprendre qu'il s'agit d'une erreur de frappe, mais malheureusement la complexité liée à l'acceptation de commandes partiellement erronnées est telle qu'il est plus simple de ne pas accepter de commandes incorrectes.

Aussi dans le cas précédent, il faut refrapper la commande :

« CHARGER MATH A PARTIR DE LA BANDE »

La commande est alors acceptée et l'ordinateur peut alors répondre : « RECHERCHE DE MATH EN COURS ».

Si l'on s'était aperçu immédiatement de l'erreur de frappe, on aurait pu utiliser les fonctions « d'édition de texte » associées au terminal. Ainsi, en appuyant sur la touche « — », le curseur se déplace d'un caractère vers la gauche. Dans ce cas, en se déplaçant en arrière jusqu'à la lettre erronnée, on peut soit la remplacer soit en insérer une autre. En appuyant sur la touche « — » plusieurs fois, on peut se repositionner à la fin du texte et continuer.

Un terminal de visualisation doit en effet permettre au minimum le déplacement du curseur dans toutes les directions, de manière à pouvoir corriger les erreurs de frappe.

Revenons maintenant à l'enregistreur de cassette.

Si l'on appuie sur le bouton « AVANCE RAPIDE », la bande se met à se dérouler. Ceci ne serait pas nécessaire avec des périphériques tels que des « disques durs » qui sont en rotation permanente.

On arrête la bande à la valeur 550 du compteur et on voit apparaître les mots suivants sur l'écran :

MORPION
ECHECS
NIM

Le microordinateur cherche alors le programme sur la bande et finalement on voit apparaître :

« MATH TROUVE »

et après un délai très court on voit apparaître :

PRET-SPECIFIER :
1 - POUR MULTIPLICATION
2 - POUR DIVISION
3 - POUR ADDITION
4 - POUR SOUSTRACTION
ENTRER VOTRE CHOIX :

La raison pour laquelle il y a eu une attente quand on arrête la bande à 550 est que, du fait de la nature séquentielle du support bande magnétique, il a fallu lire et faire défiler un certain nombre de programmes qui précédaient le programme recherché. Si l'on n'avait pas de compteur cela pourrait même prendre plusieurs minutes avant de trouver le programme recherché.

Maintenant le programme est en mémoire et on peut essayer de l'utiliser. Cette fois le programme commence par visualiser un « MENU » offrant quatre possibilités : multiplication, division, addition, soustraction. Nous tapons « 1 ».

**Question :** Est-ce suffisant ?
**Réponse :** Non. Il faut également frapper le « retour chariot » pour indiquer à l'ordinateur que l'entrée est terminée ! Si l'on a fait une erreur, on peut la corriger en faisant reculer le curseur d'une position vers la gauche. Le « retour chariot » permet donc de confirmer que l'entrée est (jugée) correcte.

Le programme répond en posant une multiplication sur l'écran :

```
   12
 X 23
 ------
 = ...?
```

et l'entrée demandée doit être la réponse qui sera vérifiée par le programme.

L'opération de ce programme doit maintenant être claire. Faisons le point.

## RECAPITULATION

On a maintenant utilisé la plupart des fonctions du système :

— le clavier a été utilisé pour entrer des commandes ou des données. Ses possibilités d'édition ont été utilisées pour corriger des erreurs en déplaçant le curseur.

— l'écran a été utilisé par l'ordinateur pour communiquer avec l'opérateur. Il visualise uniquement des caractères et non pas des images comme un appareil de télévision ordinaire.

— l'enregistreur de cassettes sert au stockage des programmes. C'est un organe bon marché, mais lent. Dans le cas de jeux et de petits programmes ceci est acceptable.

— le microordinateur possède un programme spécialisé et résident, le moniteur, qui scrute les entrées en provenance du clavier et manipule des chaînes de caractères pour déterminer s'il s'agit de commandes pour la machine. De plus, ce système dispose d'un « système de fichiers » qui permet la recherche de fichiers programmes sur la cassette (cherche « MATH »). Finalement, le microordinateur dispose d'une mémoire interne où il stocke :

1 - le programme lu à partir de la cassette.

2 - les données qui sont tapées au clavier.

Nous allons maintenant introduire l'ensemble des définitions qui nous serons nécessaires pour décrire de façon précise les composants d'un système et leurs fonctions.

# CHAPITRE III
# DEFINITIONS FONDAMENTALES

## APPLICATIONS DES MICROORDINATEURS

Il n'est pas possible de limiter en principe le nombre des applications possibles des microordinateurs. Cependant, dans le cadre de ce livre, il est pratique de classer les applications des microordinateurs en trois grandes catégories : les applications individuelles, les applications professionnelles, et les applications industrielles.

Les applications individuelles des microordinateurs sont caractérisées par des systèmes de faible coût utilisés dans un environnement domestique ou individuel. Ces systèmes sont destinés à réaliser des jeux ou des programmes divers conçus par des amateurs. Leurs caractéristiques techniques sont sensiblement différentes de celles des microordinateurs destinés aux autres applications.

Les microordinateurs de gestion sont utilisés dans les bureaux. Ils peuvent être utilisés pour des tâches commerciales telles que la gestion de la paye, la maintenance d'inventaire, la facturation. Les spécifications de ces microordinateurs en font la catégorie de système la plus coûteuse pour les trois types d'applications. Ceci est dû non seulement aux capacités de traitements demandées mais également à l'adjonction de terminaux coûteux pour obtenir les performances requises dans un environnement commercial.

Les microordinateurs industriels sont utilisés essentiellement pour le contrôle de processus. Les microcalculateurs intégrés dans un standard téléphonique, dans les ascenseurs, les voitures, les appareils médicaux font tous partie de cette catégorie.

De tels systèmes ne seront pas présentés ici. Pour une présentation détaillée des systèmes industriels, voir le livre de référence C4 (« Les microprocesseurs »).

Dans le but de caractériser les applications, il est nécessaire

d'introduire un certain nombre de termes techniques. Nous recommandons au lecteur non technicien de lire les définitions de ces termes s'il veut pouvoir comprendre les commentaires et recommandations qui seront présentés dans les chapitres suivants.

## DEFINITIONS FONDAMENTALES

Un microordinateur est un ordinateur (ou « calculateur ») dont l'unité centrale est réalisée autour d'un circuit microprocesseur en un seul boîtier.

Un microordinateur est donc un ordinateur réalisé avec des composants LSI à très haute intégration.

« LSI » (Large Scale Integration) signifie que l'on utilise des circuits à très haute intégration (grand nombre de transistors par boîtier). Cette technologie résulte de l'évolution de l'électronique digitale vers des composants de plus en plus microminiaturisés. Depuis la découverte du transistor discret bipolaire introduit après la seconde guerre mondiale, on arrive maintenant à réaliser des circuits contenant jusqu'à 50 000 transistors ou plus sur une seule pastille de silicium. La pastille (« chip ») est un petit rectangle de silicium de quelques mm de côté sur lequel a été réalisé un circuit électronique.

La technologie a progressé depuis les circuits SSI à faible intégration (SSI : small scale integration), puis MSI (medium scale integration : intégration à échelle moyenne) jusqu'aux circuits SLSI (super large scale integration) dont les densités peuvent atteindre jusqu'à 100 000 transistors par pastille et plus.

Il est maintenant nécessaire de présenter la structure de base d'un ordinateur afin de définir les éléments fonctionnels qui composent un microordinateur.

Un ordinateur est une machine qui permet d'effectuer des traitements (calculs ou manipulations de données) en exécutant un « programme ».

Un programme est une séquence d'instructions. Les instructions sont des ordres destinés à manipuler des données contenues dans un organe de mémoire. Il existe des *instructions arithmétiques* telles que l'addition, la soustraction (ou la multiplication et la division). Il existe également des *instructions logiques* permettant de réaliser des opérations telles que le OU et le ET logiques. De plus, il existe des instructions de test et de branchement qui permettent à différentes parties d'un programme de s'exécuter en fonction des conditions

testées. Le concept de programme sera étudié plus loin. Le programme est contenu dans une mémoire.

*Tout ordinateur comprend trois éléments fonctionnels de base :*

*1 - L'Unité Centrale (UC)*

Elle est chargée de rechercher les instructions contenues en mémoire et de les exécuter. Elle contient également une mémoire interne appelée « registres ». Ils sont destinés à contenir les opérations ou les résultats des opérations à effectuer. Ceci permet d'accroître la vitesse d'exécution des programmes (l'accès aux registres étant très rapide).

*2 - La mémoire centrale* utilisée pour stocker les *programmes et les données* en cours d'exécution. Il existe différents types de mémoires qui seront décrits plus loin.

*3 - Les organes d'entrées-sorties*

Ces organes permettent à l'ordinateur de communiquer avec le monde extérieur. Un organe d'entrée typique est le clavier qui permet de rentrer les données. Un organe de sortie typique est une imprimante ou un écran de visualisation sur lequel les résultats peuvent être respectivement imprimés ou visualisés.

L'ensemble des composants matériels est appelé le matériel (« hardware »). L'ensemble des programmes est appelé logiciel (« software »). Un autre terme intermédiaire appelé « firmware » est quelquefois utilisé pour des programmes contenus dans des mémoires mortes (« ROM = read only memory ») qui ne peuvent pas être modifiées. (En français « firmware » se traduit en général par « microprogrammes »).

Un *microprocesseur* est une unité centrale d'ordinateur réalisée sur un seul circuit. Le mot « chip » que l'on peut traduire par « puce » ou « pastille » est souvent utilisé pour indiquer non seulement le morceau de silicium mais également le circuit lui-même (le boîtier).

Il s'agit d'une définition quelque peu simplifiée car souvent les premiers microprocesseurs étaient en fait des unités centrales incapables de fonctionner sans un certain nombre de circuits annexes tels que : un oscillateur externe plus un quartz réalisant une horloge, des circuits conducteurs et amplificateurs de bus (« drivers »). Les *bus* sont des chemins de communication entre les différents organes d'un système. En plus des trois éléments définis ci-dessus il faut bien sûr rajouter une alimentation qui fournit les tensions nécessaires aux différents circuits. Si la mémoire du

système est petite, un système microordinateur complet à l'exclusion de l'alimentation peut être réalisé sur une seule carte électronique. Cependant dans la plupart des cas il est souhaitable de disposer d'un coffret permettant de contenir l'ensemble et de permettre des extensions ultérieures.

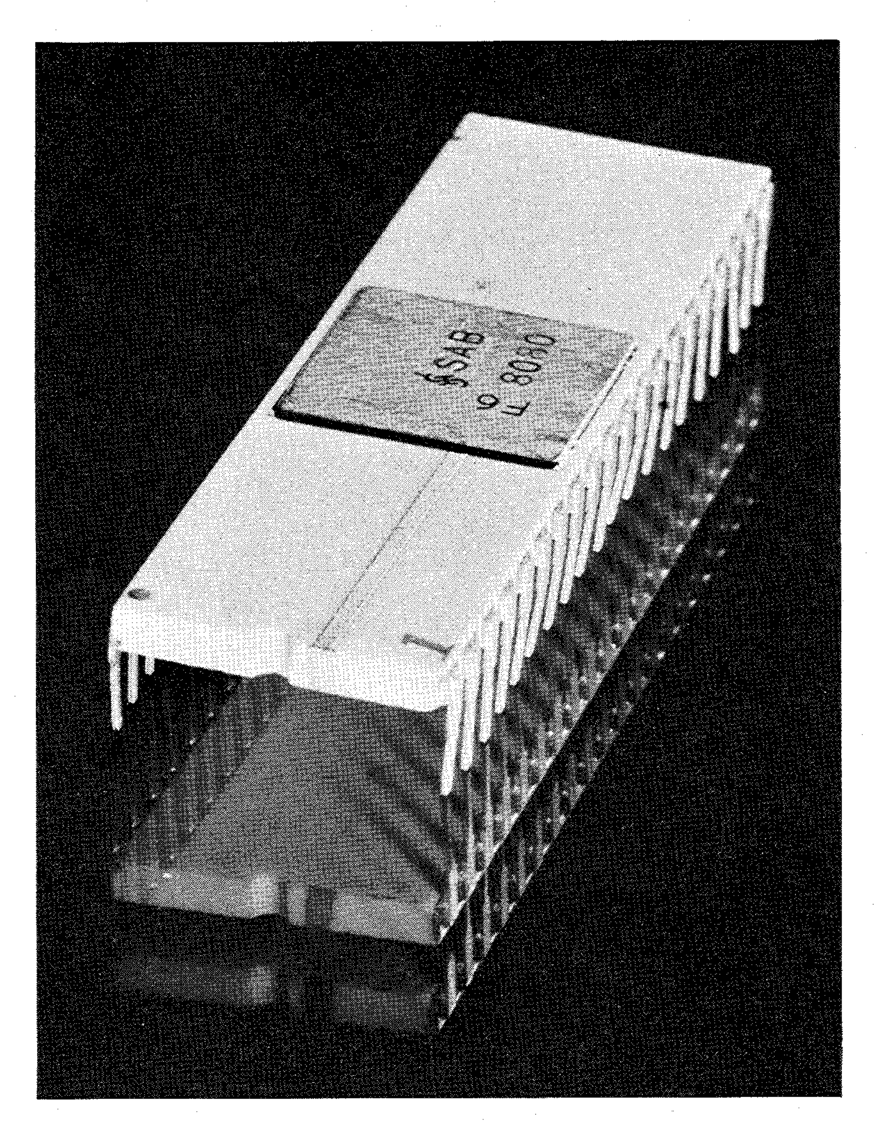



Fig. 3.1 : Un microprocesseur (8080). Le circuit possède 40 broches.

Des organes externes complexes nécessitent en plus des cartes d'interface ou des « contrôleurs ».

En général, un microordinateur a l'apparence d'une petite boîte (les dimensions dépendent du constructeur mais sont en général de la dimension d'un tiroir). Une telle boîte contient la carte unité centrale, plus une ou plusieurs cartes mémoire et des cartes d'entrées-sorties fournissant des interfaces vers des périphériques standard. Elle contient également l'alimentation.

Contrairement aux ordinateurs traditionnels, beaucoup de microordinateurs ne disposent pas de panneau avant. Un panneau avant est en général équipé de voyants lumineux et d'interrupteurs permettant de tester ou de modifier le contenu des registres ou de la mémoire de façon manuelle. On verra plus loin qu'un tel dispositif n'est pas indispensable sur un système d'ordinateur personnel ou de gestion. De ce fait la réalisation d'un panneau qui augmenterait le coût du système n'est en général pas envisagée sur les petits systèmes.

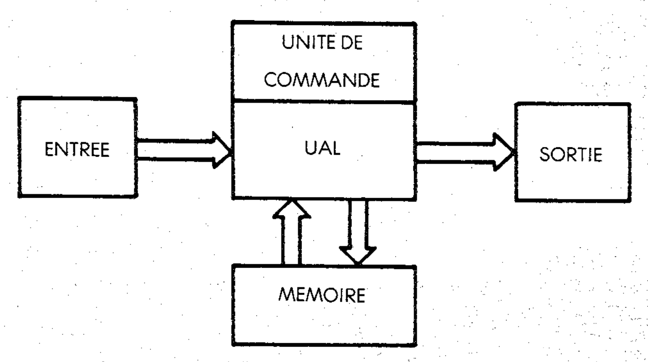


Fig. 3.2 : Les cinq éléments fonctionnels de tout ordinateur

En résumé, un microordinateur de base est un appareil contenant deux ou trois cartes logiques ainsi qu'une alimentation.

Deux périphériques standards sont généralement utilisés avec tout microordinateur. L'organe d'entrée est un clavier analogue à un clavier de machine à écrire. Chaque fois que l'utilisateur enfonce une touche de clavier, le caractère est encodé sous une forme binaire (suite de « 0 » et de « 1 »), et transmis au microordinateur.

L'organe de sortie est en général une unité de visualisation. L'unité de visualisation est un poste TV sans les dispositifs de réglage de fréquence et de réception. Il est simplement destiné à

visualiser sur l'écran les caractères engendrés par le microordinateur.

Du fait que le programme et les données nécessitent des mémoires de stockage importantes, supérieures à la capacité de la mémoire centrale, il est le plus souvent nécessaire d'adjoindre au système des organes de mémoire secondaires. Il existe deux possibilités. L'organe le moins coûteux est l'enregistreur de cassettes. Un magnétophone ordinaire peut être utilisé pour stocker des données ou des programmes sous forme digitale. C'est très bon marché mais les performances du système sont limitées.

L'autre alternative est l'unité de disque souple (« floppy disk ») qui sera décrite plus loin. Cet organe est devenu la mémoire de masse standard pour les microordinateurs. C'est un organe plus coûteux qu'un enregistreur de cassette, mais il permet un accès rapide et une capacité de stockage importante pour un coût raisonnable. Il faut cependant noter que pour toute application nécessitant la manipulation de fichiers, il est nécessaire d'utiliser *deux* unités de disques souples. De cette façon plusieurs fichiers peuvent être comparés, recopiés ou fusionnés. Ces concepts seront étudiés plus loin.

Bien sûr d'autres périphériques peuvent être rajoutés suivant les applications. Cependant, pour l'instant, nous supposerons qu'un système microordinateur standard consiste en une « boîte » microordinateur associée à un clavier (organe d'entrée), une unité de visualisation (organe de sortie) et une unité de deux disques souples (mémoire de masse), (ou une cassette). Il s'agit d'une configuration minimum et en pratique il sera le plus souvent nécessaire de rajouter une imprimante. On verra que des systèmes plus petits peuvent être envisagés pour des applications spécifiques ou individuelles. De même dans certains cas il sera nécessaire d'envisager des systèmes plus grands pour des applications plus complexes.

## La fabrication d'un microprocesseur

Comment est fabriqué un microprocesseur ?

On a vu qu'un microprocesseur est un exemple particulier de composant LSI. Un tel composant est réalisé sur une pastille encastrée dans un boîtier (« DIP » = « dual in line package ») à broches. Un tel circuit apparait sur la fig. 3.4. Il dispose de 18 à 40 broches par l'intermédiaire desquelles la pastille peut communiquer avec le monde extérieur.

Cette technologie est le résultat de l'évolution de la technologie




Fig. 3.3 : Le système comprend de gauche à droite :
1. Le microcalculateur, 2. Le disque, 3. Le terminal d'entrée-sortie

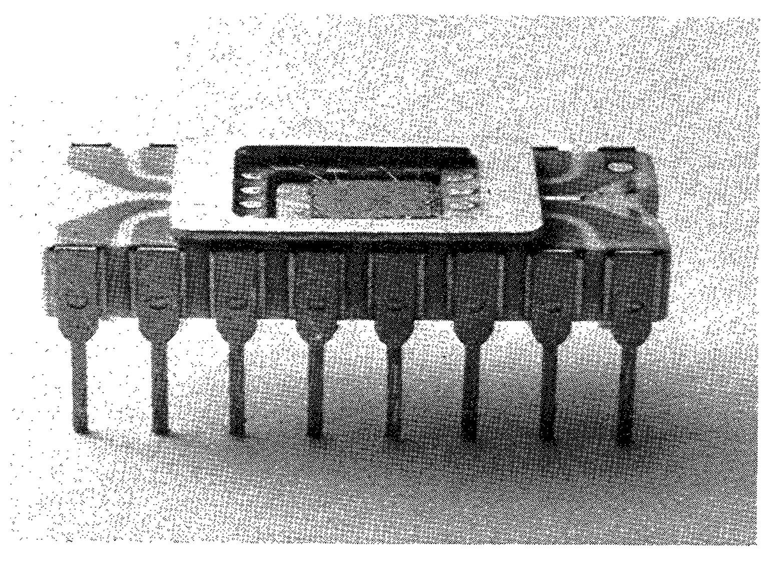


Fig. 3.4: Ce boîtier a été ouvert pour montrer la pastille («chip»).

MOS (metal oxide semiconductor). Le semiconducteur utilisé est le silicium qui est dopé avec des impuretés (phosphore ou bore) qui deviendront des charges positives ou négatives (des déficiences en électrons donneront des « trous » positifs, des excès d'électrons donneront des charges négatives). L'oxyde est l'oxyde de silicium, utilisé comme isolant sur le substrat de silicium. Le métal est la grille du transistor qui est réalisé en déposant du silicium ou de l'aluminium au-dessus de l'oxyde.

Le processus de fabrication est simple dans son principe. Un cristal de silicium pur est fabriqué avec grand soin ce qui donne un réseau cristallin parfait. Ce cylindre (de 10 à 20 cm de diamètre) est alors découpé en tranches (« wafers »). Ces tranches ou « galettes » sont polies et ont l'aspect de miroirs circulaires. Elles sont si minces qu'elles sont très fragiles et cassent comme du verre. Sur chaque tranche, on pourra créer de 100 à 500 pastilles.

Une pastille mesure environ 5 x 5 mm. Les transistors et autres composants sont créés en diffusant des impuretés dans le silicium et en déposant des couches d'oxyde et de métal. Un processus d'impression sélectif est utilisé pour définir les zones où l'oxydation ou la métallisation doivent être réalisées. Le processus est analogue à celui de l'impression d'images dans un livre. Il s'agit d'un processus photolithographique. Un masque est imprimé sur la galette de silicium et l'émulsion photographique est exposée à la lumière, développée et révélée chimiquement. La diffusion, l'oxydation et la métallisation sera réalisée sur les zones révélées sur le silicium. Quand toutes les pastilles ont été créées sur la galette de silicium, celle-ci est découpée en pastilles individuelles. Chacune de ces pastilles sera alors montée sur un boîtier qui sera scellé et deviendra un « circuit intégré ».

Après le circuit mémoire, le microprocesseur a été l'un des premiers composants généraux LSI introduit (1971). Depuis, un grand nombre de composants LSI ont été fabriqués et pratiquement tous les éléments nécessaires pour réaliser un microordinateur complet sont actuellement disponibles sous forme de circuits LSI. Du fait du grand nombre de transistors qui peuvent être implantés sur une seule pastille, il est désormais possible de réaliser un ordinateur complet sur une seule carte. Il est même devenu possible de réaliser un ordinateur complet sur une seule pastille. Cependant les microordinateurs en un seul boîtier sont limités quant à la quantité de mémoire qu'ils peuvent contenir et au nombre de lignes d'entrées-sorties. De ce fait ils sont réservés aux applications de

contrôle de processus. Dans l'avenir des cartes microordinateurs complets pourront être réalisées en un seul boîtier.

Nous allons maintenant examiner avec plus de détails chacun des trois éléments d'un système microordinateur. Il est important de comprendre la fonction de chacun des éléments pour comprendre la nécessité de tous ces composants pour la réalisation du système complet. On verra qu'il y a peu de solutions optimales, indépendantes de l'application considérée. Comme dans la plupart des systèmes complexes, le système microordinateur doit être adapté à l'application considérée. Pour qu'un tel choix soit fait de manière efficace il est nécessaire de connaître les fonctions des différents éléments d'un système. Ceci est indispensable pour évaluer les compromis techniques et économiques susceptibles d'être pris en considération. L'utilisateur commercial doit désormais être capable de sélectionner un tel système par lui-même.

## L'Unité Centrale

L'Unité Centrale (« CPU = Central Processing Unit ») décode et exécute les instructions. Elle va chercher les instructions dans la mémoire centrale et les exécute à l'aide de l'unité arithmétique et logique (ALU) qui effectue toutes les opérations arithmétiques et logiques. L'unité arithmétique et logique (UAL) est équipée de registres internes qui fournissent une mémoire rapide pour les données en cours de traitement. Les instructions sont décodées et séquencées par une unité de commande. Toutes ces fonctions sont réalisées sur une seule pastille : le microprocesseur. En pratique quelques composants supplémentaires sont généralement nécessaires : ainsi l'horloge et le quartz fournissent une fréquence de référence stable. Des composants supplémentaires sont aussi nécessaires pour amplifier les signaux : ce sont les conducteurs de bus (« drivers »). La plupart des microprocesseurs sont capables de traiter des données de 8 bits (un bit est un digit binaire qui peut prendre la valeur logique « 0 » ou « 1 »). Un microprocesseur 8 bits est un microprocesseur qui peut traiter un octet de 8 bits de données simultanément (son UAL a 8 bits de large).

## Les Bus

Le microprocesseur reçoit et transmet des données en provenance du monde extérieur par l'intermédiaire de 8 lignes de données. Ces

lignes de données constituent ce que l'on appelle le bus de données. Un bus est simplement un ensemble de lignes groupées par fonction. Le bus de données est bidirectionnel car l'information peut circuler dans les deux sens vers le microprocesseur ou en provenance du microprocesseur. Dans le but de spécifier l'origine ou la destination d'une donnée, il est nécessaire de fournir un numéro d'identification spécifié par le microprocesseur et indiquant l'emplacement de la donnée. C'est ce qu'on appelle l'adresse. De façon classique, le bus d'adresses dispose de 16 lignes. Il permet d'adresser un nombre de positions mémoire qui est égal à $2^{16} = 65\,536$ octets.

Un bus de données et un bus d'adresses ne sont pas suffisants pour avoir un système complètement opérationnel. Du fait de la nature des différents composants électroniques d'un système, il est en effet nécessaire de fournir aussi des signaux de synchronisation. Ces signaux sont transportés par un bus de commande. C'est le troisième bus connecté à un microprocesseur.

Ayant défini les bus, nous allons les utiliser, et examiner maintenant les autres composants principaux du système.

**La Mémoire**

On a vu que la mémoire d'un système stocke les programmes et les données sur lesquels ils opèrent. Une mémoire est organisée en mots. Dans un système à 8 bits les mots sont de 8 bits : dans ce cas on les appelle des octets. Dans un système à 4 bits les mots sont de 4 bits (les mots sont alors appelés « quartets »). Un mot ne spécifie donc pas un nombre de bits identique dans tous les cas. Un mot est une unité logique d'information sur laquelle le processeur travaille. Dans un système à microprocesseur 8 bits, le mot est donc égal à un octet. C'est pour cela que la taille des mémoires de microordinateurs est généralement spécifiée en octets (« byte » en anglais). Un exemple d'octet stocké dans la mémoire d'un ordinateur est : « 00000000 ». On verra dans la section programmation, que tous les caractères, toutes les données et toutes les instructions sont représentées sous forme de mots de 8 bits. Dans certains cas, il est nécessaire de prévoir plusieurs mots pour stocker une information. Du fait de l'utilisation de composants électroniques, la représentation binaire associée à des états logiques « 0 » à « 1 » est la seule qui soit compatible directement avec le matériel. Aussi cette représentation est-elle utilisée sur tous les ordinateurs. Une étude plus détaillée des programmes et des données sera présentée dans le chapitre sur la

programmation. De façon idéale la mémoire d'un système doit être rapide, de très grande capacité, et peu coûteuse. Malheureusement, la vitesse et la taille ne sont pas compatibles avec un faible coût. C'est pour cela que l'on distingue deux catégories de mémoires : la *mémoire centrale* et la *mémoire auxiliaire* ou mémoire de masse. La mémoire centrale d'un système est contenue dans le coffret du microordinateur. Elle est réalisée en composants MOS LSI.

La mémoire de masse est généralement une mémoire à support magnétique tel que disque souple ou cassette magnétique. Ce type de mémoire est beaucoup moins coûteux et offre des capacités de stockage beaucoup plus importantes que la mémoire centrale, mais il est plus lent.

Les différents types de composants utilisés pour la réalisation de la mémoire centrale seront décrits dans le chapitre suivant. Deux types essentiels sont utilisés : les mémoires mortes (ROM) et les mémoires vives (RAM) à accès aléatoire.

Malheureusement une mémoire vive réalisée en technologie MOS est volatile. Son contenu disparaît lorsque le circuit n'est plus alimenté. Pour cette raison un programme permanent doit être stocké dans des mémoires mortes (« ROM = read only memory ») qui ne peuvent pas être réécrites et ne sont pas volatiles.

Ces deux types de mémoires (ROM ou RAM) sont toujours nécessaires sur un système microordinateur. Dans le cas de systèmes professionnels ou individuels, les programmes du constructeur résident sur des ROM et les programmes utilisateurs résident sur des RAM. Quand les programmes utilisateurs ne sont pas exécutés, ils sont conservés sur des disques ou des cassettes.

On verra que la taille mémoire nécessaire dépend de l'application considérée. Une taille typique est de 2 à 8 K de ROM et au moins 8 K de RAM pour un « petit système », et 40 à 48 K de RAM pour un système à usage professionnel.

## Les Entrées-Sorties

Le périphérique d'entrée universel est le clavier alphanumérique. Le clavier alphanumérique est un clavier similaire à celui d'une machine à écrire équipé d'un certain nombre de caractères spéciaux supplémentaires. Le clavier est associé à un encodeur qui envoie directement des codes de 7 ou 8 bits vers le microprocesseur à chaque fois qu'une touche est enfoncée.

Il s'agit du périphérique le moins coûteux pour un opérateur

humain. Dans certains environnements spécialisés d'autres mécanismes d'entrée peuvent être utilisés. Ainsi pour les jeux télévisés des manches à balai (« joy-stick ») ou des manettes spécialisées peuvent être utilisés pour jouer. Dans ce cas ils peuvent être meilleur marché mais sont plus limités en capacité de transmission d'information.

Les deux périphériques de sortie universels pour un microordinateur sont l'unité de visualisation et l'imprimante. Il n'est pas indispensable d'avoir les deux et une imprimante peut être suffisante. Cependant une imprimante fiable et assez rapide est bien plus coûteuse qu'une unité de visualisation qui permet d'examiner de longs textes assez rapidement. Une unité de visualisation (écran TV) est donc utilisée avec pratiquement tous les systèmes microordinateurs car elle permet de visualiser rapidement et de façon silencieuse de grandes quantités d'information. Dans les cas où une trace permanente est nécessaire, il faudra donc rajouter une imprimante.

Du fait que les organes d'entrées-sorties sont pour la plupart des organes électro-mécaniques relativement complexes chaque appareil nécessite un *contrôleur* qui est une carte électronique assurant l'interface entre le microprocesseur et le périphérique (décodage des commandes et lecture des états du périphérique). Ces contrôleurs assurent également la lecture et/ou l'écriture des données encodées dans un format associé au microprocesseur.

Dans la plupart des cas, il n'est donc pas possible de connecter directement un microordinateur à un organe d'entrée-sortie. Des circuits contrôleurs ou d'inteface sont nécessaires. Dans le cas des périphériques simples, l'interface fait généralement partie du boîtier microordinateur, ou bien est incorporé dans le périphérique. C'est ainsi que nous avons connecté notre clavier et notre terminal de visualisation directement par un câble.

## Un système à microprocesseur

Un système à microprocesseur est une carte électronique complète à l'exception de l'alimentation, des coffrets et des périphériques. L'architecture d'un tel système est présentée sur la fig. 3.5. Sur la gauche, apparaît le microprocesseur qui est l'unité centrale du système. Sur la droite apparaissent les deux types de mémoires : ROM et RAM. Enfin à l'extrême droite, apparaît le circuit d'interface d'entrée-sortie qui permet une communication

directe avec des périphériques. Les longues bandes horizontales sur le schéma représentent les trois bus du système.

Sur la gauche apparaissent l'horloge et le cristal qui sont externes à l'unité centrale. L'interface entrée-sortie crée un bus d'entrée-sortie qui est connecté directement aux périphériques. De nombreux efforts de standardisation ont été faits pour fournir une connection facile à des périphériques externes. On verra, en particulier que l'un des standards les plus utilisés dans le marché des amateurs est le bus S100 de manière interne, et le bus RS232 de manière externe. Ils seront mentionnés plus loin.

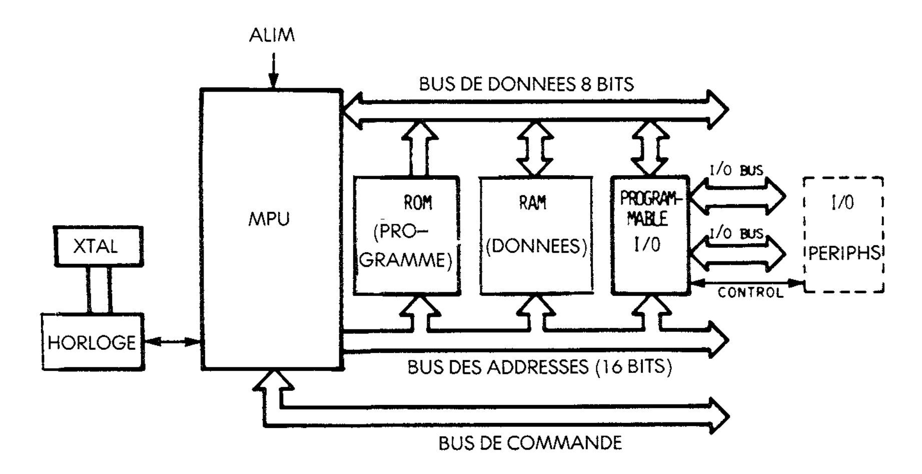


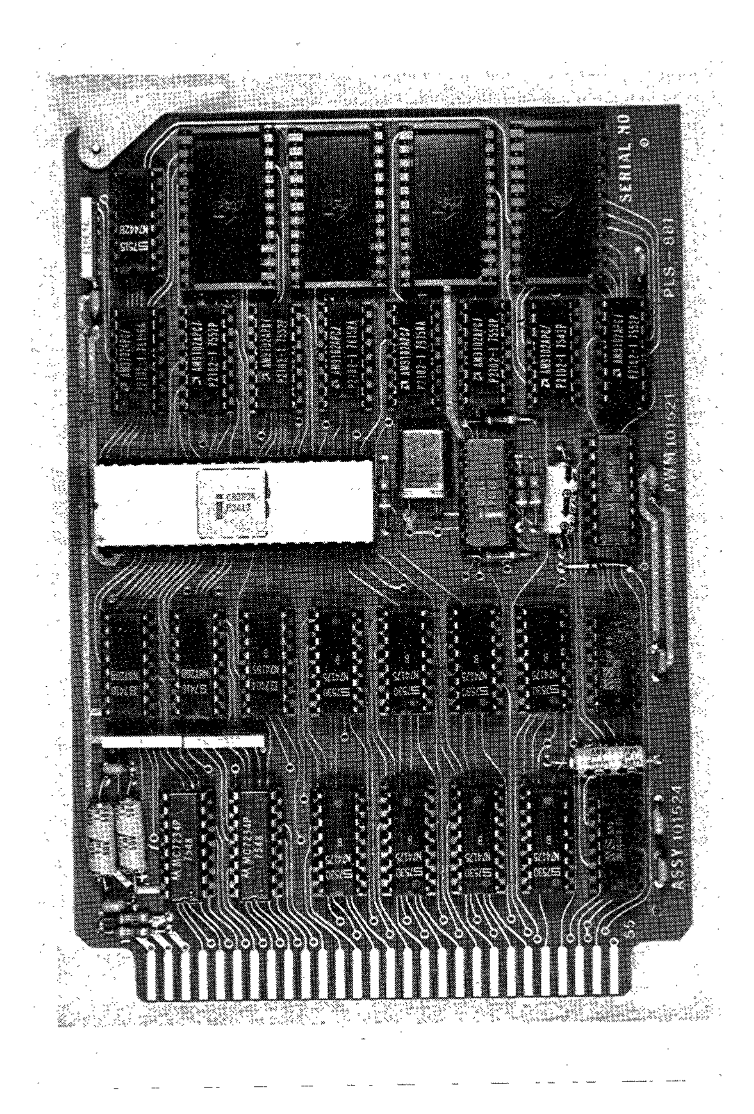

Fig. 3.5 : Système à microprocesseur standard

## Le système microordinateur :

Un système microordinateur est un système complet et directement utilisable construit autour d'une carte à microprocesseur. Il inclut tous les éléments fonctionnels décrits ci-dessus. En particulier il comprend au minimum : une boîte microordinateur (contenant une ou plusieurs cartes et l'alimentation), un clavier, une unité de visualisation et/ou une imprimante.

## Microordinateur ou Miniordinateur ?

Il a toujours été difficile de donner une bonne définition de ce qu'est un miniordinateur. En général, ce terme évoque une machine




3.8 : Microordinateur sur une carte : le microprocesseur est le grand
au centre.

signaux simultanément. La taille de mot est de 16 bits le plus souvent et l'exécution des instructions est plus rapide.

De ce fait un miniordinateur est encore notablement plus rapide qu'un microordinateur moyen. Cependant des microprocesseurs 16 bits sont actuellement disponibles et leur puissance devient similaire à la plupart des miniordinateurs.

Une autre différence importante réside dans le fait que du fait de leur existence sur le marché depuis plus de 10 ans, les miniordinateurs disposent maintenant de bibliothèques de programmes et de logiciels que n'ont pas encore les microordinateurs. Cependant ce handicap est en train de se combler rapidement grâce au marché des amateurs. Ce point sera développé plus loin. Pour des applications scientifiques et/ou temps réel complexes, un miniordinateur est encore souhaitable. Cependant, pour un grand nombre de nouvelles applications qui ne demandent pas une puissance de calcul importante, le microordinateur est suffisant.

Remarquons que ce n'est pas une panacée. Un microordinateur ne peut pas traiter toutes les applications. Il existe encore de sérieuses limitations quant à l'utilisation de microordinateurs dans le domaine commercial.

Des exemples de miniordinateurs qui sont devenus des microordinateurs sont : le NOVA/2 de Data General, ou le LSI 11 de Digital (utilisé dans le PDP 11/03) ; ils utilisent maintenant des circuits microprocesseurs.

## Avantages des microordinateurs

Les avantages des microordinateurs sont ceux associés à la technologie LSI. Ce sont essentiellement :

### 1. La taille réduite

Un système microordinateur complet peut être réalisé dans un volume très réduit. Il existe même des microordinateurs en un seul boîtier qui peuvent être utilisés lorsque le besoin en mémoire est limité. Un système complet dont la capacité de traitement est comparable à celle d'un miniordinateur peut être réalisé dans un volume correspondant à un attaché-case. Dans certains systèmes le microordinateur peut même être incorporé directement dans le clavier du terminal !

**2. Le coût très faible**

Le coût d'un boîtier MOS est de quelques dizaines de francs. Un système complet nécessite un nombre relativement faible de circuits LSI ce qui donne des prix de revient très faibles pour un microordinateur complet. Le prix dépend de la taille mémoire requise et des périphériques associés, il est actuellement possible d'acquérir un système complet pour quelques milliers de francs.

**3. La souplesse de la programmation**

Un microordinateur réalise par programme ce qui pouvait l'être par des moyens mécaniques, manuels ou autres.

L'avantage de la programmation est la grande souplesse que cela introduit pour modifier, tester ou améliorer un système. Ceci sera plus clair lorsque nous aurons parlé de programmation.

## CONCLUSION

Toutes les définitions de base ont été introduites. Des exercices sont présentés à la fin de ce chapitre pour tester l'assimilation de ces notions de base. Il est indispensable de bien comprendre ces définitions pour pouvoir comprendre les chapitres suivants.

Du fait du grand nombre d'options, chaque système diffère dans la taille mémoire, les unités d'entrées sorties, etc. Cependant tous les systèmes qui seront présentés ont la même organisation basée sur un petit nombre d'éléments fonctionnels. Leur évaluation nécessite une compréhension de leur fonction.

Pour le lecteur plus intéressé par la technique, le chapitre suivant va décrire le fonctionnement interne des différents éléments. Il peut être sauté par les autres.

# EXERCICES

*(Réponses à la page suivante)*

3.1 Lequel des éléments suivants réside dans un coffret de microcalculateur ?
1. la carte unité centrale. 2. les cartes de mémoire additionnelle. 3. la carte de contrôleur du disque. 4. la carte d'interface spécialisée. 5. l'alimentation. 6. le décodeur de clavier.

3.2 Quelle est la différence entre un miniordinateur et un microordinateur ?

3.3 Définir un bus.

3.4 Quelle est la différence entre une ROM et une RAM ?

3.5 Est-ce que tout système nécessite une RAM ?

3.6 Est-ce que la mémoire de masse est indispensable ?

3.7 Quels sont les trois éléments fonctionnels de tout ordinateur ?

3.8 Combien de bits y a t-il dans un octet ?

3.9 Combien de bits y a t-il dans un mot ?

3.10 Définir une adresse.

3.11 Est-ce que des caractères sont des données ?
Est-ce que des nombres sont des données ?

## REPONSES

3.1 1-2-3-4-5.
Le décodeur de clavier est sous le clavier. Naturellement si le clavier est intégré au coffret du microordinateur, il se trouve également dans le coffret.

3.2 Voir le texte.

3.3 Voir le texte.

3.4 Voir le texte.

3.5 Oui, sinon aucune entrée/sortie ne pourrait être réalisée, car les données doivent être stockées dans des RAM.

3.6 Non. Si les programmes sont courts ou s'ils sont stockés dans des ROM.

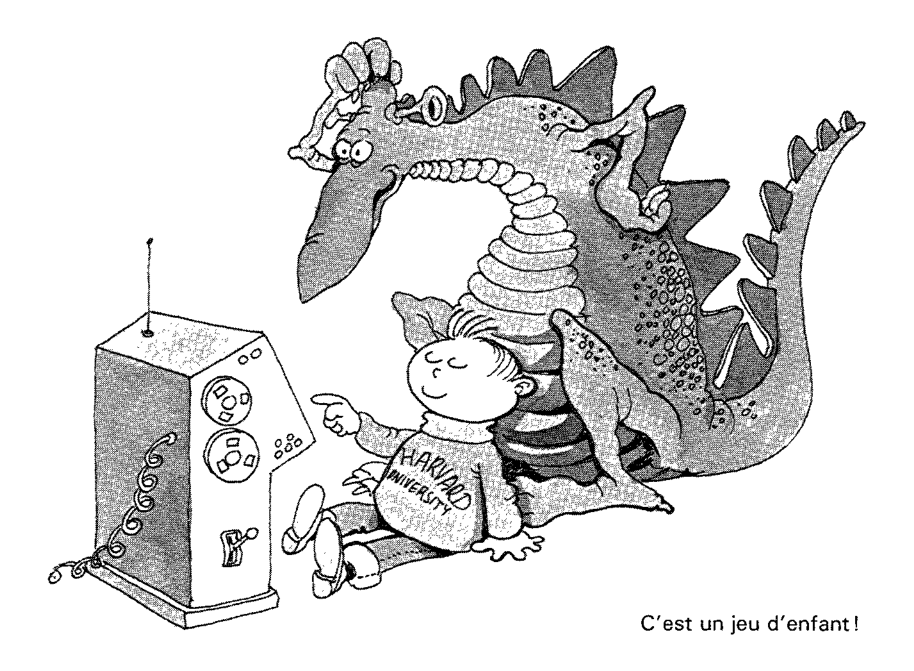
3.7 Unité centrale, Mémoire centrale, Entrées/Sorties.

3.8 8 bits.

3.9 Non défini. Dépend du processeur.

3.10 Une adresse est un nombre qui correspond à l'endroit où une donnée peut être trouvée.

3.11 Oui.
Oui.




C'est un jeu d'enfant!

# CHAPITRE IV
# COMMENT IL FONCTIONNE

## INTRODUCTION

Le but de ce chapitre est d'expliquer d'une manière plus détaillée les fonctions de chaque composant d'un système microordinateur. Cette information technique est nécessaire pour ceux qui désirent évaluer les capacités techniques d'un système et ses limitations pour une application donnée. Il est aussi nécessaire pour l'utilisateur qui désire développer une application commerciale et s'intéresse à la façon dont cela fonctionne.

Cependant ce chapitre peut être sauté pour tous ceux qui ne sont pas intéressés par les aspects techniques. Le chapitre précédent était simplement une introduction de base. Ce chapitre est plus technique et fournira au lecteur intéressé une compréhension plus détaillée de la façon dont les différents composants fonctionnent.

## L'ARCHITECTURE DE BASE D'UN SYSTEME

Les trois éléments fonctionnels de base de tout ordinateur sont : l'unité centrale, la mémoire et les entrées-sorties.

Comme il n'est pas encore possible de réaliser un nombre de transistors suffisants sur une seule pastille, chacune de ces fonctions est réalisée par un ou plusieurs boîtiers. Ces trois fonctions et leur réalisation à l'aide de composants spécifiques vont être étudiées en détail ci-dessous.

### L'unité centrale

L'*unité centrale* (CPU,UC) de l'ordinateur est chargée de rechercher, décoder et exécuter les instructions. Celles-ci sont contenues dans la *mémoire* du système. Un exemple pratique montrera comment une instruction est recherchée en mémoire, amenée dans l'unité centrale, décodée et exécutée.

Pour des raisons de commodité, on va supposer ici que l'UC est réalisée à l'aide d'un boîtier microprocesseur (appelé habituellement MPU : microprocessor unit). Ce n'est pas tout-à-fait vrai : dans la plupart des systèmes, une horloge, un quartz et, quelquefois des

composants extérieurs supplémentaires peuvent être nécessaires. Cela suffira pour notre exemple. La figure 4.1 montre le microprocesseur sur la gauche et la mémoire à droite.

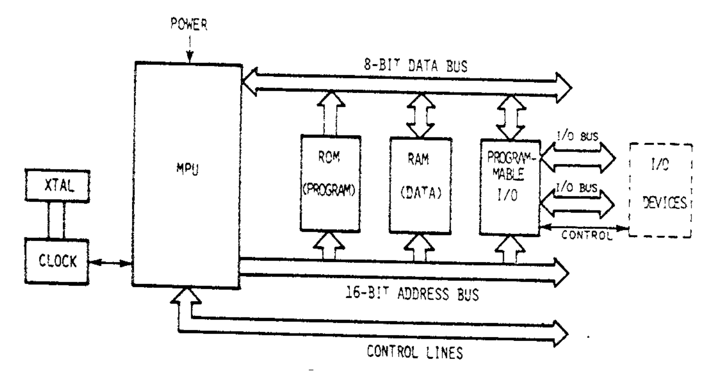



Fig. 4.1 : Système standard.

Un programme est exécuté instruction par instruction. Normalement ces instructions sont exécutées en séquence et sont contenues dans la mémoire. Dans le but de mémoriser l'emplacement d'une instruction en mémoire, l'adresse de la prochaine instruction à exécuter est contenue dans un registre interne du microprocesseur appelé le compteur ordinal ou compteur de programme. (PC en anglais).

Un *registre* est une mémoire interne au microprocesseur et peut contenir un ou plusieurs mots d'information. Dans notre exemple on supposera que le compteur ordinal est un registre de 16 bits c'est-à-dire qu'il peut contenir 16 digits binaires (0 ou 1). Ces 16 bits représentent l'adresse binaire de la prochaine instruction à aller rechercher en mémoire. La recherche de cette instruction est alors simple : il suffit de déposer le contenu du registre compteur de programme sur le bus d'adresse du microprocesseur. En effet notre microprocesseur standard possède 16 broches d'adresses qui permettent la propagation d'une adresse vers l'extérieur. Ces 16 lignes constituent le bus d'adresse du système. Comme une mémoire peut en général être lue ou écrite, un signal de lecture sera nécessaire. Le microprocesseur engendrera ce signal et la mémoire lira le contenu d'une adresse spécifique. Ce contenu constituera la prochaine instruction qui sera exécutée par le microprocesseur. Quelques centaines de nanosecondes ($10^{-9}$ seconde) s'écouleront avant que la donnée soit disponible dans la mémoire.

Ce délai est appelé le *temps d'accès* de la mémoire. Dans un système microordinateur standard la mémoire est composée de mots de 8 bits (octets). En réponse à une adresse de 16 bits, il y a recherche d'un mot de 8 bits dans la mémoire. Il est important de souligner qu'il n'y a aucune relation entre le nombre de bits fournis sur le bus adresse et le nombre de bits fournis à la sortie de la mémoire.

Un exemple similaire est fourni par une adresse postale. La donnée de cette adresse permet de retrouver une maison ou un bâtiment mais le fait que l'adresse de la rue soit petite ou grande est indépendant de la taille du bâtiment qui se trouve à cette adresse.

L'instruction de 8 bits doit maintenant être acheminée vers le microprocesseur afin d'être exécutée. Toutes les données du système transitent par l'intermédiaire d'un bus spécial appelé bus de données. La mémoire est équipée de 8 connections avec le bus de données. Les 8 bits de l'instruction sont envoyés sur le bus de données que l'on voit apparaître en haut à droite sur la figure 4.1.

La donnée transite sur ce bus en direction du microprocesseur. Elle est envoyée dans un registre interne du microprocesseur que l'on appelle le registre d'instruction (RI). Ce registre est un registre de 8 bits et contient donc l'instruction qui va être exécutée. Quand le moment arrive d'exécuter cette instruction, ces bits sont envoyés vers un décodeur interne et des signaux de commande appropriés sont engendrés par le microprocesseur, ce qui résulte en l'exécution de l'instruction.

Le mécanisme de recherche et d'exécution d'une instruction doit maintenant être clair.

Pour rechercher la prochaine instruction il existe un mécanisme automatique : le compteur de programme est associé à un incrémenteur (opération « + 1 »). Cet opérateur incrémente le contenu du compteur de programme à la fin de chaque cycle d'exécution. De cette façon, à chaque fois que le contenu du compteur de programme est envoyé sur le bus d'adresses il y a un calcul automatique de l'adresse de l'instruction suivante dans la mémoire. Ceci constitue un mécanisme de séquencement automatique qui permet d'exécuter les instructions contenues dans des adresses successives de la mémoire.

Bien sûr, il existe des cas où l'exécution d'un programme nécessite la recherche d'une instruction qui ne se trouve pas en séquence. Ceci est appelé une instruction de *branchement*. Dans ce cas, une instruction spéciale de branchement modifiera explicite-

ment le contenu du compteur ordinal de manière à forcer le processeur à « sauter » à une adresse différente. Ceci sera vu dans la section programmation.

Une question peut encore se poser à l'esprit : comment une instruction est-elle physiquement exécutée dans l'unité centrale ?

Une description simple sera donnée ici. Pour une description plus complète voir le livre C4 « Les microprocesseurs ».

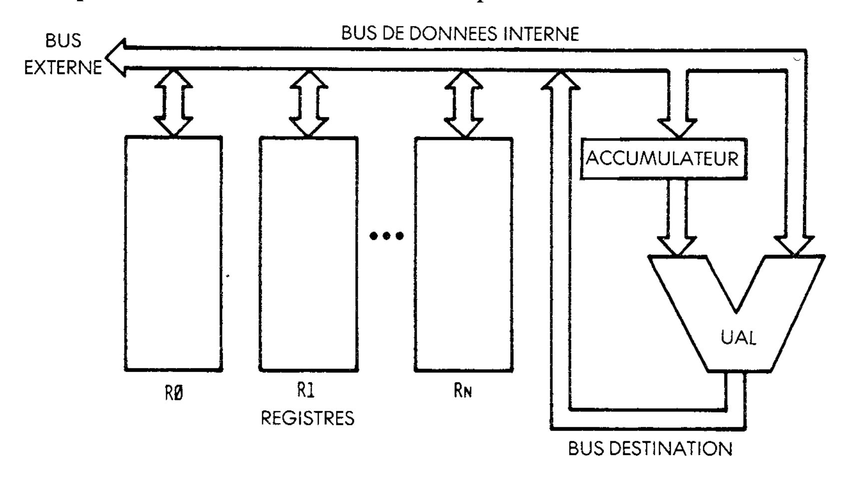



Fig. 4.2 : L'intérieur du microprocesseur

La recherche et le décodage d'une instruction est effectuée par *l'unité de contrôle* (ou de « commande ») de l'unité centrale. L'exécution de l'instruction est réalisée par l'unité arithmétique et logique (UAL) qui dispose de registres internes. Ceci est représenté sur la figure 4.2. Les rectangles verticaux sur la gauche de la figure sont des registres. Dans le cas d'un microprocesseur « standard » ces registres sont de 8 bits. Le symbole en V représente l'unité arithmétique et logique qui est chargée d'exécuter les opérations arithmétiques et logiques spécifiées par une instruction. De plus l'UAL est équipée d'un décaleur qui permet de décaler une donnée vers la droite ou vers la gauche. Associé à cette UAL, existe aussi un registre d'état qui contient des bits d'états. Un bit d'état est un bit qui est utilisé pour mémoriser une condition interne au microprocesseur. De telles conditions peuvent être : une retenue arithmétique, un résultat nul, un résultat négatif ou tout autre évènement notable susceptible d'être mémorisé. Des instructions spéciales du pro-

gramme pourront tester ces conditions et causer des branchements en fonction des évènements détectés.

En vue d'illustrer la façon dont une unité centrale fonctionne, nous allons décrire l'exécution d'une addition du contenu de deux registres internes. On supposera que les registres R0 et R1 contiennent des données de 8 bits correspondant aux operandes de l'addition. Le but est d'additionner le contenu de R0 et R1 et de stocker le résultat dans R0. A la fin de l'opération le contenu de R1 devra être intact et R0 contiendra la somme des deux nombres contenus dans R0 et R1. L'instruction d'addition entre registres est exécutée en plusieurs temps.

Le contenu de R0 est envoyé sur le bus de données interne (en haut de la figure 4.2) vers l'entrée de droite de l'UAL. Puis le contenu de R1 est envoyé par le même bus vers l'entrée de gauche de l'UAL. Puis l'ordre d'addition est transmis par l'unité de commande à l'UAL. L'addition est effectuée et le résultat est envoyé par les lignes de sorties de l'UAL sur le bus (en bas de la figure 4.2). Ce résultat est alors dirigé vers le registre de destination R0 par le bus interne de données.

Comme la lecture d'un registre ne modifie pas son contenu, le registre R1 sera inchangé après l'exécution de l'instruction d'addition. Par contre R0 contiendra la somme des deux nombres précédemment contenus dans R0 et R1.

On notera qu'à la suite de l'addition, un ou plusieurs bits d'états seront positionnés (en particulier la retenue et le signe du résultat).

Le but de ce livre n'est pas de détailler plus encore le fonctionnement de l'unité centrale. Le lecteur intéressé est encouragé à consulter nos livres de référence pour des informations plus détaillées.

Nous allons résumer le fonctionnement de l'unité centrale : l'unité de commande recherche une instruction dans la mémoire, l'amène dans le registre d'instruction, la décode et engendre les signaux de contrôle nécessaire à l'exécution automatique. L'UAL est chargée d'exécuter des opérations soit à partir de registres internes à l'unité centrale soit directement à partir de données venant de l'extérieur via le bus de données en provenance de la mémoire ou d'organes d'entrée-sortie.

Nous allons maintenant étudier plus en détail le monde extérieur au microprocesseur : la mémoire et les organes d'entrée-sortie.

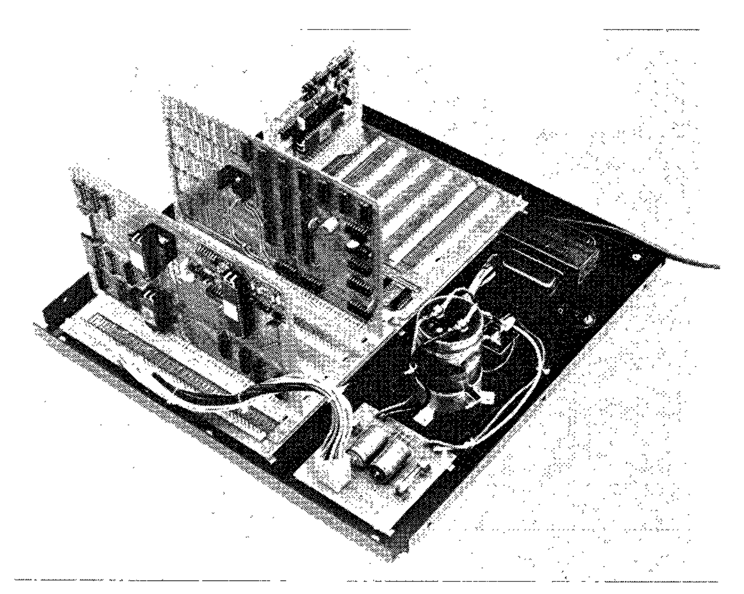


Fig. 4.3 : Intérieur d'un microordinateur (SWTPC). Les trois cartes de gauche sont l'unité centrale, la mémoire, l'interface d'entrée-sortie. L'alimentation se trouve sur la droite.

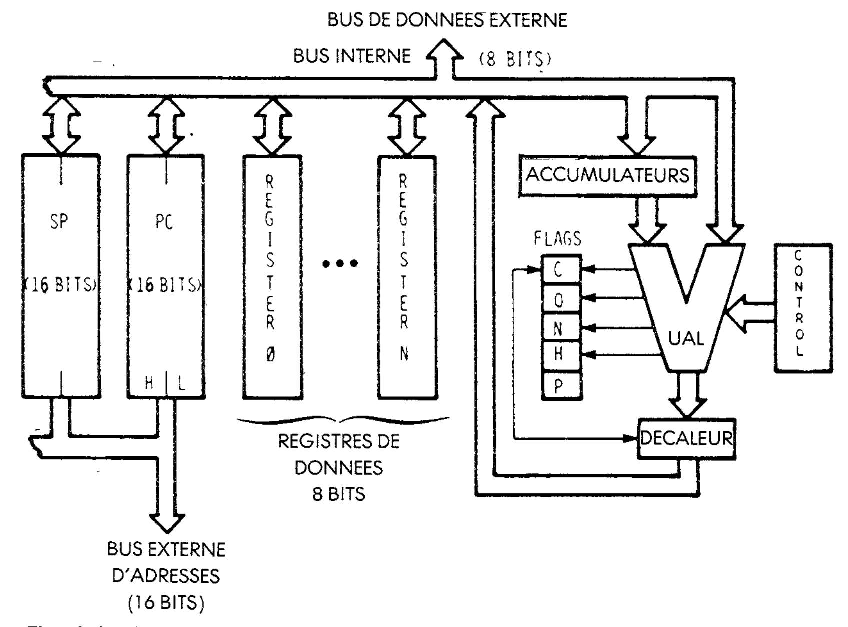



Fig. 4.4 : Un diagramme plus détaillé permet de voir les indicateurs, le circuit de décalage et le pointeur de programme.

**La mémoire**

Nous avons vu que la mémoire stocke le *programme*. La mémoire doit aussi conserver les *données* que l'utilisateur a entrées, ou qu'il crée durant l'exécution d'un programme. Nous avons souligné que la mémoire MOS (celle qui atilise des circuits intégrés) a un inconvénient notable en ce moment : les mémoires vives, c'est-à-dire les mémoires où l'information peut être soit lue soit écrite (emmagasinée) sont *volatiles : leur contenu disparaît lorsqu'on débranche la prise de courant.*

Pour cette raison, deux types de mémoires doivent être utilisés dans les systèmes microordinateurs : les ROMs et les RAMs. Dans les systèmes plus anciens, tels que les miniordinateurs, ce problème ne se posait pas. En effet, la technologie des mémoires à tores était utilisée. Chaque bit d'information (c'est-à-dire un « 0 » ou un « 1 » logique) était mémorisé dans un tore de ferrite, en forme d'anneau circulaire, et comme ce tore était toujours magnétisé cette mémorisation était permanente même en l'absence d'alimentation. L'introduction de la technologie MOS a rendu nécessaire l'utilisation de deux types de mémoires : les RAM et les ROM.

Le premier type de mémoire appelé RAM (random access memory) est une mémoire à accès aléatoire qui peut être écrite ou lue. Les RAM doivent être utilisées dans un système pour stocker les données sinon il ne serait pas possible pour l'unité centrale de mémoriser les résultats de calculs ou des données en provenance d'un clavier ou de tout autre organe d'entrée. La taille d'un programme, ou fichier, que l'on veut pouvoir charger dans un système est donc limitée par la quantité de RAM dont le système dispose. On verra dans la section suivante que deux types de technologies sont utilisés pour réaliser des RAM : les mémoires statiques ou dynamiques.

Le deuxième type de mémoire est appelé ROM (Read only memory) ou mémoire morte. Il s'agit de mémoire non volatile utilisée pour stocker des programmes. Une fois l'information stockée dans une ROM, elle ne peut pas (en principe) être modifiée. Lorsqu'un programme est correct, il ne sera plus modifié et peut par conséquent résider dans une ROM. Ceci est le cas de la plupart des applications industrielles. En particulier le programme moniteur de votre microordinateur est normalement en ROM car il n'est pas modifié par l'utilisateur. Cependant la plupart des utilisateurs qui développent un programme eux-mêmes veulent avoir la possibilité

de le modifier. Pour cette raison, différents types de ROM sont disponibles sous forme de PROM ou RPROM. Une PROM est une mémoire morte programmable par l'utilisateur. Elle utilise une technologie avec fusibles qui peuvent être claqués ou non (représentant ainsi « 0 » ou « 1 ») par un appareil relativement peu coûteux appelé programmateur de PROM.

Les PROM sont des circuits bon marché mais du fait qu'elles utilisent une technologie avec fusible, elles ne peuvent plus être modifiées une fois programmées. C'est pour cela qu'ont été introduits les circuits RPROM et EPROM qui sont des circuits de mémoires mortes reprogrammables. Les RPROM ou EPROM sont effaçables par rayonnement ultra-violet et les EAROM sont effaçables électriquement. Avec ce type de mémoire, l'utilisateur peut programmer le circuit, exécuter un programme et plus tard en effacer le contenu et le reprogrammer. Ce type de mémoire est normalement utilisé pour tous les programmes en développement.

Cependant, les applications individuelles et commerciales diffèrent des applications industrielles. En effet dans ce type d'applications, un grand nombre de programmes différents doivent être exécutés dans la journée et doivent être successivement rentrés dans la mémoire. Pour cette raison, des mémoires mortes ne sont pas utilisées dans ces systèmes sauf pour la partie résidente du moniteur. Par contre les programmes utilisateurs doivent être chargés dans des

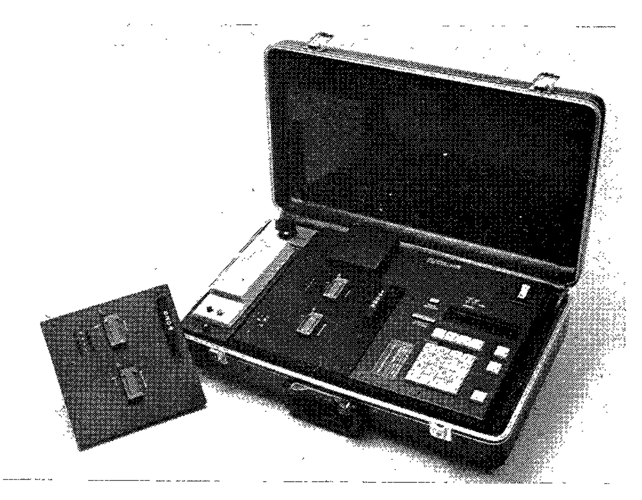


Fig. 4.5 : Programmateur des PROM dans un attaché case.
Les supports permettent d'insérer les PROM, le motif binaire est introduit à l'aide du clavier hexadécimal.

mémoires RAM. On sait que dans ce cas, à chaque coupure de tension, le programme disparaît de la mémoire. Ceci explique que de tels systèmes doivent être équipés d'une mémoire de masse auxiliaire telle que disque souple sur laquelle les programmes sont stockés de façon permanente. A chaque fois que l'utilisateur veut exécuter un programme, il doit être recopié en RAM à partir du disque. Ceci est une opération simple, rapide, qui peut être faite automatiquement si le système dispose d'un DOS (« disk operating system ») c'est-à-dire d'un logiciel de fonctionnement avec disque. Ces opérations sont alors transparentes à l'utilisateur.

**Les RAM statiques et dynamiques**

Le détail des technologies ne sera pas abordé ici. Il faut savoir simplement que pour les systèmes ayant des mémoires de faible capacité, des circuits RAM statiques seront utilisés car cela est moins coûteux globalement. Pour des systèmes ayant des mémoires plus importantes (au-dessus de 16 K octets), les RAM dynamiques sont utilisées. Pour les systèmes à application commerciale ou individuelle il y faudra généralement plus de 16 K de mémoire et les mémoires utilisées seront plutôt dynamiques. Cependant si cela est moins coûteux, des cartes mémoires statiques pourront être utilisées sans qu'il y ait de différence au niveau de l'utilisateur. Par conséquent nous ne rentrerons pas dans le détail des compromis à trouver entre ces différentes technologies. (Ceci est expliqué dans le livre de référence C4).

**Les organes d'Entrées-Sorties**

Les organes d'entrée-sortie les plus courants sont le clavier pour l'entrée et le tube cathodique pour la sortie. D'autres appareils peuvent être utilisés. En général ces appareils fournissent ou acquièrent des données de 8 bits en parallèle. Ainsi un clavier standard enverra sur le bus de données les 8 bits correspondant au code du caractère associé à la touche enfoncée.

De la même manière lorsqu'un caractère est envoyé à l'unité de visualisation elle l'est sous forme de 8 bits en parallèle. D'autres organes d'entrée-sortie utilisent un processus de communication série. Un Télétype par exemple est une machine à écrire avec clavier et mécanisme d'impression. Quand une touche est enfoncée, une succession de 11 bits est envoyée vers le système microcalculateur.

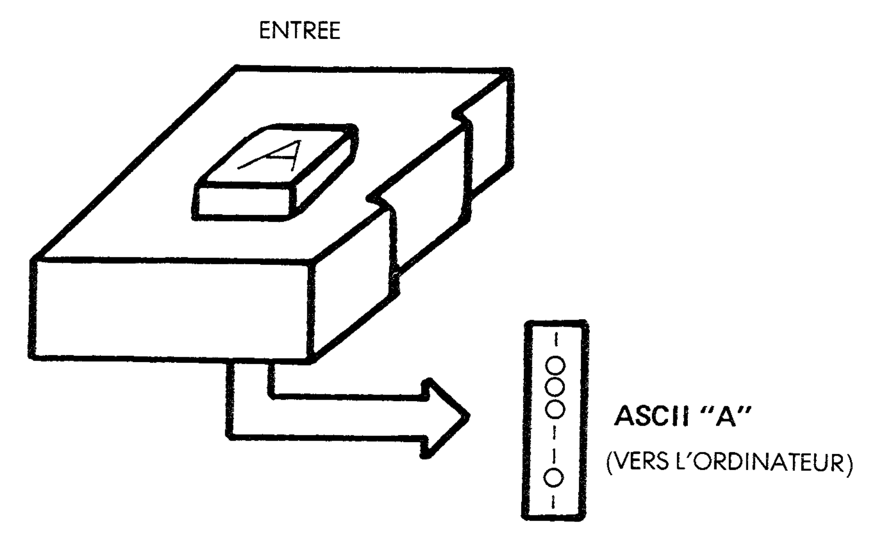



Fig. 4.6 : L'appui d'une touche de clavier engendre un code 8 bit, transmis en parallèle

Le premier bit est appelé le bit de départ, il est suivi de 8 bits de données qui correspondent au code du caractère, et le message est terminé par 2 bits d'arrêt. Comme dans le cas précédent, les 8 bits représentent l'information associée à un caractère mais cette fois les bits sont transmis en série les uns après les autres.

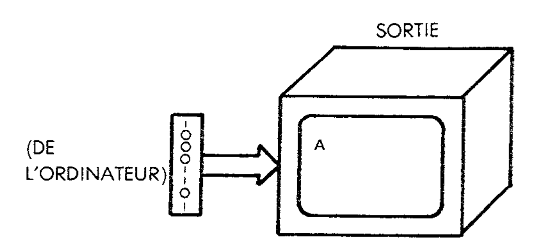



Fig. 4.7 :Le code binaire pour « A » est envoyé à l'unité de visualisation

En conclusion, il existe des organes d'entrée-sortie série ou parallèle. Deux types de circuits d'interface ont donc été créés pour assurer les connexions aux bus d'un microordinateur. Dans certains cas cependant des organes d'entrée-sortie complexes nécessitent une logique d'interface plus complexe. En particulier, dans certains cas, des codes de 8 bits envoyés par le microordinateur devront être interprêtés par le périphérique comme des commandes et non plus des données. Dans ce cas, il sera donc nécessaire d'avoir en plus un

décodeur de commandes et le tout constituera une carte appelée « contrôleur » de l'appareil d'entrée-sortie. De tels contrôleurs reçoivent des commandes, les décodent et les traduisent en une séquence de fonctions à effectuer correspondant à la commande. Ceci est nécessaire en particulier pour les terminaux de visualisation « intelligents » et les unités de disques.

Les deux circuits de base qui sont utilisés dans un système microordinateur pour réaliser des interfaces série ou parallèle sont respectivement le UART et le PIO. Les abréviations peuvent varier suivant les constructeurs mais leur fonctionnement est similaire.

Un UART (« Universal Asynchronous Receiver Transmitter ») est un « circuit universel récepteur et transmetteur asynchrone ». Le détail de son fonctionnement ne sera pas étudié ici. Le schéma général est présenté sur la fig. 4.8. Il s'agit simplement d'un convertisseur série-parallèle et parallèle-série.

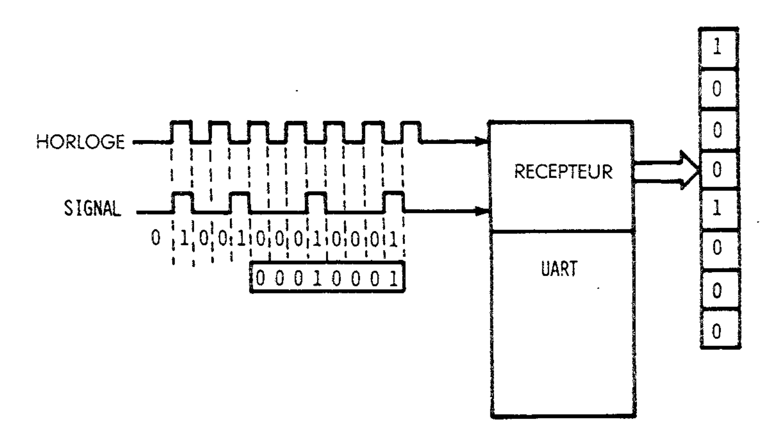



Fig. 4.8 : L'UART convertit des données série en parallèle et parallèles en série

Le rôle d'un UART est de convertir des données d'un format série en un format parallèle et inversement. Ainsi un UART peut recevoir des entrées parallèles sur 8 bits et les convertir en une séquence de 8 bits (ou plus) sur une seule ligne. Simultanément il peut recevoir des signaux en série et les convertir en parallèle. Les entrées ou sorties 8 bits en parallèle sont connectées au bus de données bidirectionnel d'un microprocesseur. Les entrées et sorties série sont connectées au périphérique (cassette par exemple).

Le circuit PIO (« Parallel programmable Input-Output ») est un circuit programmable d'échange de données en parallèle. Il est également connecté aux bus du microprocesseur et possède deux (ou plus) ports d'entrée/sortie. (Un *port d'entrée/sortie* est simplement une connexion de 8 bits en parallèle avec le monde extérieur). De plus il dispose de *registres-tampons* qui mémorisent l'information reçue ou à envoyer. Enfin, le PIO est programmable et peut gérer des protocoles de commandes d'entrée-sortie. C'est ce que l'on appelle la « poignée de main ».

Il suffit ici de dire que les circuits PIO ou UART sont des circuits universels, utilisés sur presque toutes les cartes microordinateur, qui permettent de réaliser facilement des interfaces série ou parallèle. Bien sûr, de la logique additionnelle peut être nécessaire si l'on désire non seulement transmettre des caractères mais également des commandes et des mots d'états. Dans ce cas il est nécessaire de développer une carte logique spécifique appelée contrôleur.

Les boîtiers principaux d'entrée-sortie d'un microordinateur ont maintenant été décrits. Des circuits additionnels spéciaux peuvent être nécessaires. En particulier, il existe maintenant des circuits contrôleurs en un seul boîtier tels que le circuit contrôleur de disques souples (FDC) ou le circuit contrôleur d'unité de visualisation (CRTC). Il existe également des circuits convertisseurs de signaux analogiques et de signaux digitaux. Le lecteur est renvoyé aux ouvrages C4 et C5 pour des renseignements complémentaires sur ces circuits.

## L'alimentation

Le circuit d'alimentation est chargé de fournir une tension régulière et stable aux composants. C'est un élément important d'un système, tant du point de vue coût, que du point de vue fiabilité globale du système. Du point de vue physique c'est aussi un élément important, car il occupe souvent 1/5 à 1/4 du volume total du coffret microordinateur.

Les quatre éléments d'une alimentation sont présentés sur la fig. 4.9.

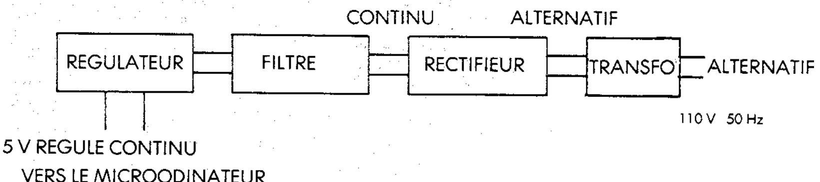



Fig. 4.9 : Les quatre éléments d'une alimentation

## RECAPITULATION

Les trois éléments fonctionnels d'un système microcalculateur ont été présentés : l'unité centrale, la mémoire et les organes d'entrée-sortie. Chacune de ces fonctions peut être réalisée par circuits LSI spécifiques. Tous ces circuits peuvent être montés sur une ou plusieurs cartes qui constituent le microordinateur. L'exécution interne d'une instruction à l'intérieur de l'unité centrale a été décrite. Nous sommes maintenant prêts à utiliser le système. Cela signifie qu'il nous faut étudier l'écriture des programmes et leur utilisation en vue des applications. Faisons-le.

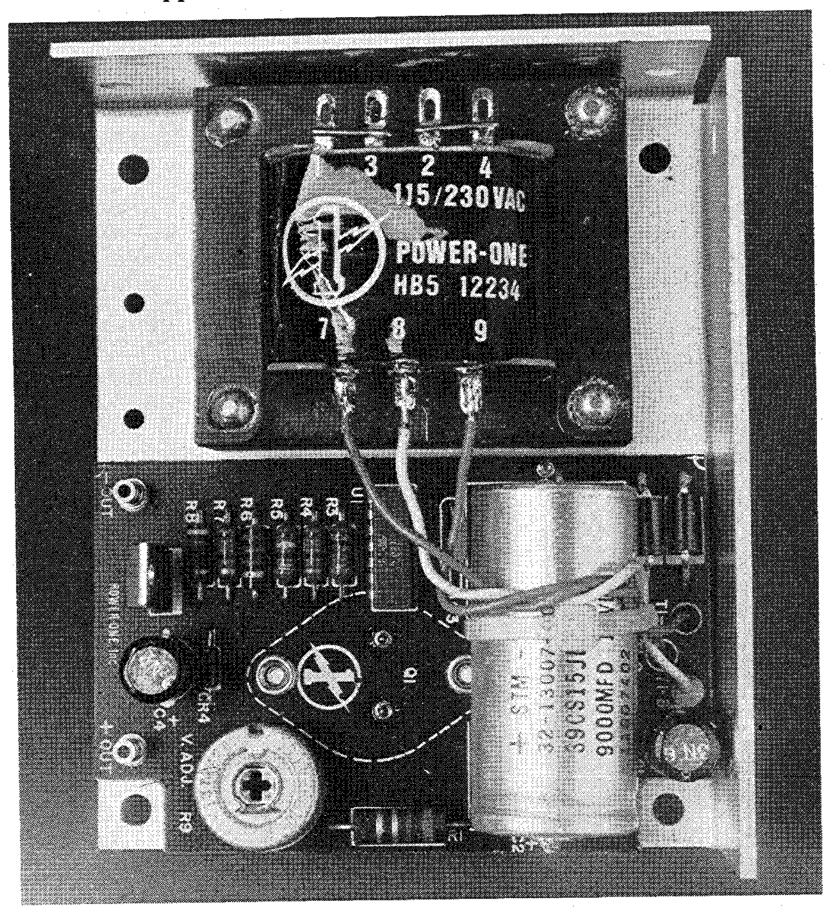
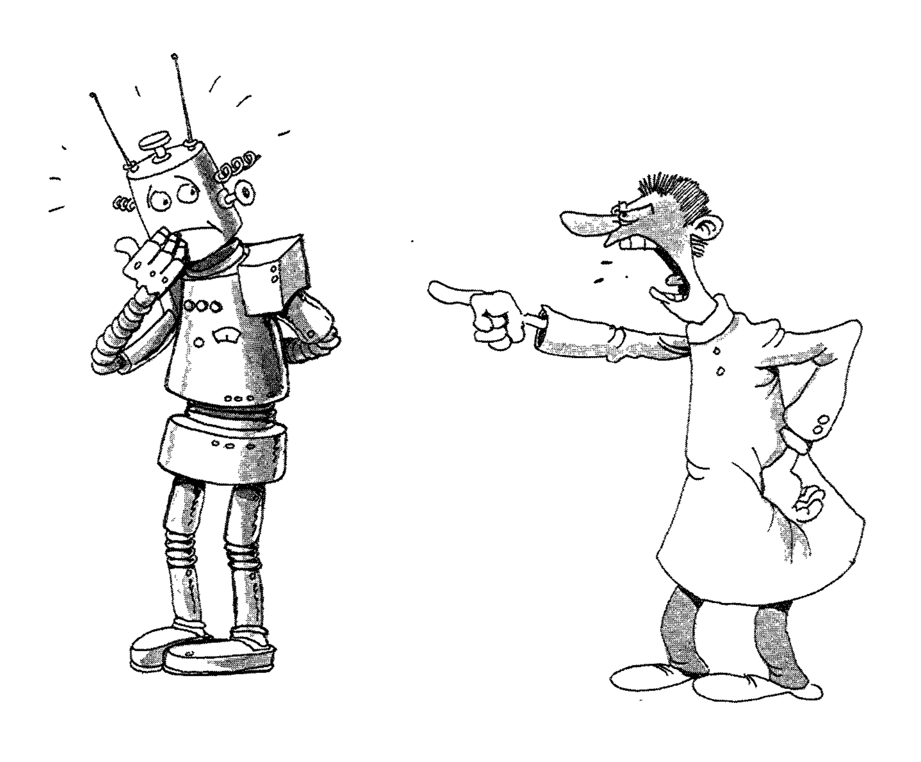



Fig. 4.10 : Une alimentation pour microordinateur montre ses éléments fonctionnels (transfo en haut, condensateur circuit)



« Je n'ai pas fait d'erreur de programmation ! »

# CHAPITRE V
# LA PROGRAMMATION

Un programme est une séquence d'ordres, dits « instructions » soumis à un ordinateur pour résoudre un problème spécifique. Dans le domaine des applications commerciales on parle de « traitement informatique des données ». Dans un tel environnement, la fonction principale d'un programme est de traiter des « fichiers » contenant des données.

Dans le domaine des applications individuelles, les programmes peuvent être beaucoup plus variés. Ils peuvent réaliser des jeux, des systèmes de contrôle ou d'alarme, ou de traitement de texte.

De façon idéale, il serait souhaitable de pouvoir dire directement en français, à l'ordinateur ce qu'il faut faire. Malheureusement, le langage naturel ne peut être utilisé pour communiquer avec la machine car il est ambigu. Seul un sous-ensemble du langage naturel peut être utilisé pour donner des commandes à l'ordinateur. D'autre part, la nécessité de traitement rapide impose que les instructions soient de longueur limitée. Pour ces raisons, des langages de programmation spécialisés ont été créés pour indiquer de façon concise et précise le traitement à effectuer. Il faut cependant se rappeler que le seul langage qui correspond à la structure de l'ordinateur est le langage binaire (des suites d'instructions représentées par des « 0 » et des « 1 »). C'est ce que l'on appelle le langage machine.

Une fois qu'un problème a été défini, sa solution sera spécifiée par un *algorithme*. Un algorithme est une spécification de la solution du problème, étape par étape. A titre d'exemple, voici un algorithme simplifié de contrôle d'un feu de carrefour :

— allumer le vert sur la voie A
— attendre 2 minutes
— éteindre le vert
— allumer le jaune sur la voie A
— attendre trente secondes

— éteindre le jaune
— allumer le rouge
— allumer le vert sur la voie B
— attendre une minute
— allumer le jaune sur la voie B
— attendre vingt secondes
— allumer le rouge sur la voie B
— retourner à la première étape.

Cet algorithme est une « boucle » de contrôle de l'intersection. Bien sûr un microordinateur pourrait être utilisé pour gérer de manière dynamique l'intersection. L'algorithme serait alors amélioré. L'exemple ci-dessus est aussi un exemple de « boucle de programme » : la même séquence d'instructions sera ici répétée indéfiniment.

La traduction d'un algorithme dans un langage de programmation constitue la programmation. Elle peut être faite au niveau du langage machine ou à un niveau dit « plus élevé ». Examinons ceci.

*1. Langage machine*

La programmation en langage machine implique des instructions spécifiques que l'ordinateur peut exécuter immédiatement. Comme il est fastidieux de programmer en langage machine (suite de « 0 » et « 1 » et que cela est source d'erreur, une représentation symbolique des instructions est utilisée. Ceci est appelé le *langage assembleur.*

Ainsi par exemple :

« ADD R0, R1 » signifie : « additionner le contenu des registres R0 et R1 et mettre le résultat dans R0. Cette instruction est une instruction symbolique. Dans le but de l'exécuter sur un microprocesseur il est nécessaire de la traduire en suite de bits. Cette traduction est réalisée par un programme spécial, appelé *assembleur.*

L'assembleur est un programme de traduction automatique du langage assembleur en codes binaires exécutables. Chaque instruction symbolique est traduite en une instruction binaire. Ces instructions binaires qui font de 8 à 24 bits sont alors stockées dans la mémoire et peuvent y être exécutées.

L'entrée d'un programme dans la mémoire du système peut être faite directement dans le cas d'une ROM en insérant le composant ou bien via un périphérique, par exemple en tapant sur le clavier, ou en lisant une cassette. Dans un système typique, le programme sera frappé au clavier, lu dans la mémoire RAM puis (à l'intérieur du

microordinateur) stocké sur une mémoire de masse telle qu'un disque. Pour utiliser le programme, il faudra plus tard le charger à partir du disque dans la RAM où il peut être traduit en binaire et exécuté.

*2. Langage de haut niveau.*

La deuxième possibilité est d'écrire un programme en langage de haut niveau. Un langage de haut niveau est plus proche du langage naturel, du moins scientifique ou de gestion. Il est plus difficile à utiliser car les instructions sont relativement faciles à comprendre mais doivent toujours obéir à une syntaxe rigide de façon à éliminer toutes les ambigüités. Il existe un grand nombre de langages de haut niveau disponibles. Cependant pour les applications de gestion ou individuelles il y a actuellement un langage qui prédomine pour les microordinateurs : c'est le langage BASIC. La programmation dans ce langage sera présentée plus loin. Un autre langage est souvent utilisé dans les applications industrielles et de contrôle c'est PL/M (« Langage de Programmation pour Microprocesseurs »), développé par le constructeur INTEL. Une des différences essentielles entre les deux langages est que le PL/M est un langage « compilé » tandis que BASIC est un langage « interprété ». Nous allons clarifier ces termes.

Un *interprêteur* est un programme qui interprète et exécute les instructions d'un langage de haut niveau tel que BASIC. Bien sûr les instructions dans le langage BASIC doivent être converties dans un format binaire pour être exécutées. Mais il existe deux possibilités de traduction : chaque instruction peut être traduite en code machine séquentiellement, créant ainsi un « code-objet » qui peut être exécuté séparément ou bien chaque instruction peut être traduite et exécutée *immédiatement après.* Quand chaque instruction est traduite immédiatement et exécutée, on dit que le langage est *interprêté.* Au contraire quand tout le programme est d'abord traduit en code objet, prêt pour une exécution future, on dit que le langage est *compilé.* Les programmes qui effectuent ces traductions sont appelés respectivement un interprêteur et un compilateur.

L'avantage d'un interprêteur est que pour chaque ligne frappée correspondant à une instruction celle-ci peut être exécutée immédiatement. Cela permet de modifier des instructions sans avoir à recompiler l'ensemble du programme. Ceci offre une plus grande flexibilité pour le développement ou la modification du programme. Le désavantage en est la perte d'efficacité au niveau de l'exécution.

Lorsque le programme comporte des boucles, il est nécessaire pour l'interprêteur de retraduire ces instructions à chaque itération. Avec un compilateur, les instructions associées seraient traduites une seule fois et l'exécution serait donc beaucoup plus rapide. Un compilateur traduit tout le programme-source en code-objet en une seule fois. Puis l'exécution a lieu à partir d'un programme objet (programme binaire). Si une ligne du programme est à modifier cela implique la retraduction de *tout le programme* et donc un délai important. C'est pour cette raison que les langages interprêtés sont préférés pour le développement des programmes, car ils assurent une interaction directe avec le système. En effet une instruction peut être exécutée, modifiée puis réexécutée. Dans le but de remédier aux moindres performances à l'exécution, des versions compilées du langage BASIC sont également disponibles. Quand un programme a été complètement testé et est prêt à être exécuté sans modification, un compilateur BASIC traduira le programme source en programme objet qui sera exécuté de façon plus performante (en principe du moins, car un bon interprêteur peut être plus rapide qu'un mauvais compilateur).

Il est important de noter ici que beaucoup d'utilisateurs obtiennent des performances suffisantes avec un interprêteur et que les avantages fournis au moment du développement sont souvent supérieures aux inconvénients liés à la perte de vitesse d'exécution due à l'interprêteur. Cela ne serait pas vrai dans un environnement de contrôle de processus et la plupart des utilisateurs dans ce cas préfèrent un langage compilé tel que PL/M ou même le langage assembleur qui permet de gagner encore en efficacité à l'exécution.

## LES LANGAGES DE PROGRAMMATION

Un grand nombre de langages de programmation ont été inventés. Chacun d'eux offre des avantages spécifiques pour une certaine catégorie d'utilisateurs. On doit se rappeler que le critère principal pour juger un langage de haut niveau est son adéquation à la résolution des problèmes de l'utilisateur. Cela veut dire d'une part, la facilité de programmation, d'autre part la disponibilité de programmes commerciaux écrits dans ce langage.

Comme PL/M a été le premier langage compilé développé sur microprocesseurs il est largement utilisé. Plusieurs versions de FORTRAN sont maintenant aussi disponibles. Il s'agit d'un langage compilé très utilisé dans le domaine de l'informatique scientifique et

il existe de grandes bibliothèques de programmes écrites dans ce langage.

Pour l'instant, le seul interprêteur disponible et très répandu sur les microprocesseurs est le BASIC. Un autre interprêteur qu'il serait souhaitable de développer est APL, mais il n'est pas encore généralement disponible sur microprocesseur. Le langage PASCAL offre aussi des avantages.

## LES ORGANIGRAMMES

La phase qui précède la programmation est de façon classique appelée « organigramme » ou « ordinogramme ».

Un organigramme est une représentation symbolique d'un algorithme.

Deux symboles de base sont utilisés : les rectangles et les losanges.

Un rectangle est utilisé pour des instructions représentant des ordres à la machine tel que « ouvrez la porte » ou « A = 2 ».

Un losange est utilisé pour représenter un test tel que « Si cela bout, ajouter le produit chimique C » ou bien « Si A = 2 aller au point 51 de l'organigramme » ou « Si X = 1 imprimer ERREUR ».

Considérons l'exemple suivant qui décrit un « algorithme de thermostat ».

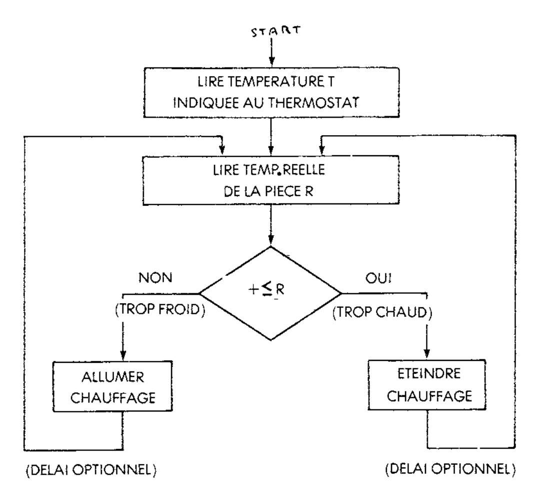



Fig. 5.1 : L'organigramme représente l'algorithme

Dans l'étape 1, la température spécifiée normalement sur le thermostat est « T ».

Dans l'étape 3, la température de la pièce P est mesurée à l'aide d'un thermomètre.

— si la pièce est trop chaude (P > T) alors on éteint le chauffage

— si la pièce est trop froide (P ⩽ T) alors on allume le chauffage.

C'est ce qui est spécifié dans les étapes 3, 4, 5.

Le losange utilisé en 3 indique qu'il y a un test. Les deux flèches à la sortie du losange indiquent les deux alternatives en fonction du résultat du test.

**QUESTION 1 :** *Si plusieurs flèches arrivent sur une étape est-ce obligatoirement un losange (test) ?*

**QUESTION 2 :** *Peut-il y avoir plus d'une flèche quittant un rectangle ?*

**REPONSES :** *1. Non — 2. Non.*

Quand un algorithme a été traduit en organigramme, il devient plus simple à programmer. La méthode traditionnelle adoptée dans l'industrie est généralement :

1. trouver l'algorithme qui résoud le problème,
2. traduction en organigramme,
3. programmation dans un langage approprié.

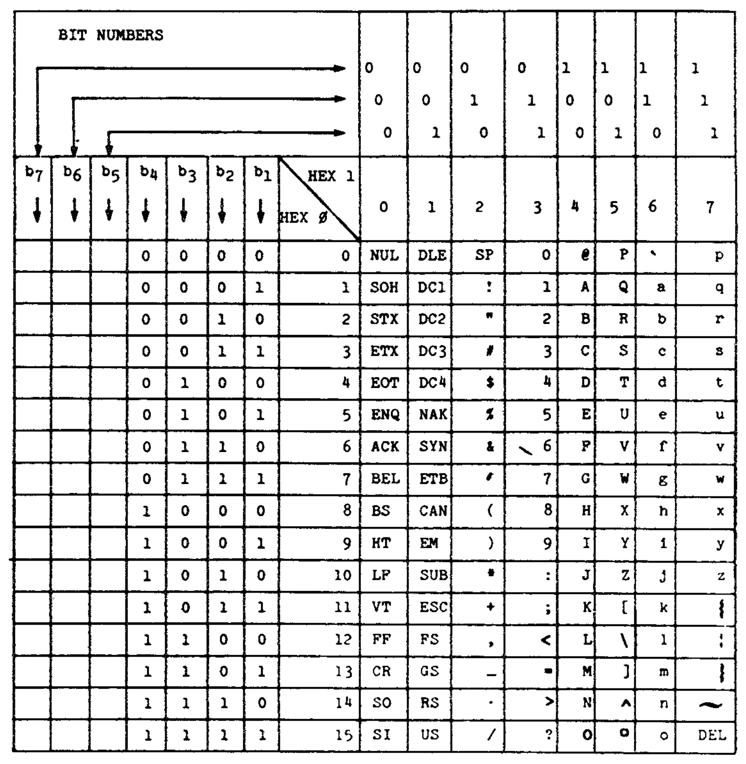
L'expérience indique qu'approximativement 10 % de la population peut programmer directement à partir d'un algorithme et obtenir un programme correct. Malheureusement, l'expérience indique aussi que 90 % de la population pense faire partie de ces 10 % !

En résumé, beaucoup de gens pensent qu'il n'est pas nécessaire de passer par la phase de l'organigramme. L'expérience montre que 90 % des gens ne le pensent pas jusqu'à ce que leur programme ne fonctionne pas. Il est donc fortement suggéré d'établir un organigramme avant d'écrire le programme. Il est important de noter que l'organigramme est facile à lire et constitue une documentation des programmes accessible à tous puisque l'on peut y utiliser des phrases de langage naturel, des abréviations, ou des instructions proches d'un langage de programmation. Il indique la séquence des opérations effectuées au cours du temps.

Des symboles spéciaux sont quelquefois utilisés pour indiquer des opérations d'entrées sorties : cartes perforées, imprimante, disque, bande magnétique. Ces symboles sont en général évidents, et nous n'en parlerons pas.

## REPRESENTATION DE L'INFORMATION DANS UN MICROORDINATEUR

Les circuits électroniques digitaux sont caractérisés par des états binaires représentant le « 0 » ou le « 1 ». tous les ordinateurs doivent donc représenter l'information sous forme binaire. Le système binaire permet de représenter facilement des nombres comme des séquences de « 0 » et de « 1 ». De façon pratique, on a vu qu'un microprocesseur 8 bits manipule des informations de 8 bits (octets). Toutes les données ou instructions seront donc sur un ou plusieurs octets. Les instructions de la plupart des microordinateurs par exemple sont soit de un, deux ou trois octets. Les caractères de l'alphabet sont codés sur un octet : en effet 8 bits permettent $2^8$ combinaisons de « 0 » et « 1 », soit 256 codes différents. On peut donc représenter 256 caractères différents (ou 128 si l'on utilise un « bit de parité ». Un exemple de code caractère très utilisé est donné pour la table du code ASCII (fig. 5.2).



| BIT NUMBERS | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $b_7$ → | | | | | | | | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| $b_6$ → | | | | | | | | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| $b_5$ → | | | | | | | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| $b_7$ | $b_6$ | $b_5$ | $b_4$ | $b_3$ | $b_2$ | $b_1$ | HEX 1 / HEX Ø | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| | | | 0 | 0 | 0 | 0 | 0 | NUL | DLE | SP | 0 | @ | P | ` | p |
| | | | 0 | 0 | 0 | 1 | 1 | SOH | DC1 | ! | 1 | A | Q | a | q |
| | | | 0 | 0 | 1 | 0 | 2 | STX | DC2 | " | 2 | B | R | b | r |
| | | | 0 | 0 | 1 | 1 | 3 | ETX | DC3 | # | 3 | C | S | c | s |
| | | | 0 | 1 | 0 | 0 | 4 | EOT | DC4 | $ | 4 | D | T | d | t |
| | | | 0 | 1 | 0 | 1 | 5 | ENQ | NAK | % | 5 | E | U | e | u |
| | | | 0 | 1 | 1 | 0 | 6 | ACK | SYN | & | 6 | F | V | f | v |
| | | | 0 | 1 | 1 | 1 | 7 | BEL | ETB | ' | 7 | G | W | g | w |
| | | | 1 | 0 | 0 | 0 | 8 | BS | CAN | ( | 8 | H | X | h | x |
| | | | 1 | 0 | 0 | 1 | 9 | HT | EM | ) | 9 | I | Y | i | y |
| | | | 1 | 0 | 1 | 0 | 10 | LF | SUB | * | : | J | Z | j | z |
| | | | 1 | 0 | 1 | 1 | 11 | VT | ESC | + | ; | K | [ | k | { |
| | | | 1 | 1 | 0 | 0 | 12 | FF | FS | , | < | L | \ | l | ¦ |
| | | | 1 | 1 | 0 | 1 | 13 | CR | GS | _ | = | M | ] | m | } |
| | | | 1 | 1 | 1 | 0 | 14 | SO | RS | . | > | N | ^ | n | ~ |
| | | | 1 | 1 | 1 | 1 | 15 | SI | US | / | ? | O | ° | o | DEL |

Fig. 5.2 : Table ASCII

| | |
|---|---|
| ACK | Acknowledge |
| BEL | Bell |
| BS | Backspace |
| CAN | Cancel |
| CR | Carriage return |
| DC1 | Direct control 1 |
| DC2 | Direct control 2 |
| DC3 | Direct control 3 |
| DC4 | Direct control 4 |
| DEL | Delete |
| DLE | Data link escape |
| EM | End of medium |
| ENQ | Enquiry |
| EOT | End of transmission |
| ESC | Escape |
| ETB | End transmission block |
| ETX | End text |
| FF | Form feed |
| FS | Form separator |
| GS | Group separator |
| HT | Horizontal tab |
| LF | Line feed |
| NAK | Negative acknowledge |
| NUL | Null |
| RS | Record separator |
| SI | Shift out |
| SOH | Start of heading |
| SP | Space |
| STX | Start text |
| SUB | Substitute |
| SYN | Synchronous idle |
| US | Unit separator |
| VT | Vertical tab |

Fig. 5.3 : Abréviations table code ASCII

La représentation de grands entiers nécessite deux ou plusieurs octets. Avec deux octets soit 16 bits on peut aller jusqu'à 64K (1K = 1024) soit 65536. Ceci n'est pas toujours suffisant et il existe des conventions spéciales de représentation des nombres décimaux telles que la « représentation en virgule flottante ». A titre d'exemple, apparaît ci-dessous la représentation binaire d'entiers de 0 à 7. Pour représenter ces 8 combinaisons soit $2^3$ il faut 3 bits.

| ENTIER | REPRESENTATION BINAIRE |
|---|---|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

Ceci permet en particulier de condenser une représentation binaire en représentation octale et de remplacer des groupes de 3 bits par le chiffre octal (de 0 à 7) correspondant.

Cette représentation a beaucoup été utilisée sur certains miniordinateurs. En pratique comme les microprocesseurs utilisent des mots de 8 bits ces mots sont souvent représentés dans une représentation *hexadécimale* (ensemble de 4 bits associés à un chiffre hexadécimal). Il s'agit d'un détail technique mineur.

## OPERATIONS LOGIQUES

L'arithmétique binaire est très simple dans le cas des entiers :

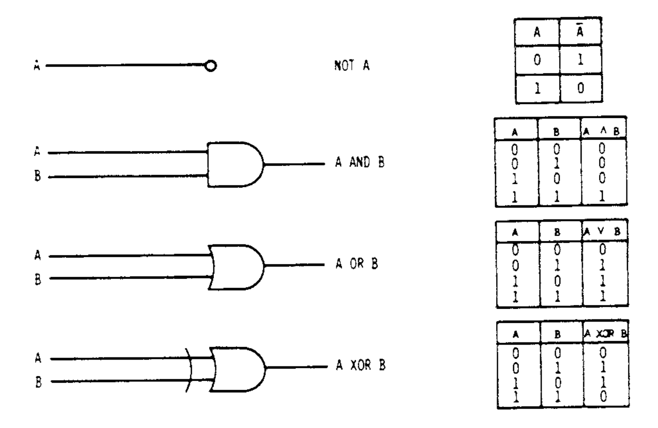
$0 + 0 = 0$

$0 + 1 = 1$

$1 + 1 = 10$ (retenue de 1).

Les quatre principales opérations logiques sur les nombres binaires sont la négation NON, OU (représenté par le symbole $V$), ET (représenté par $\Lambda$ ), le OU exclusif (XOR).

Chaque opération est définie par une « table de vérité » qui donne la valeur du résultat de l'opération logique en fonction des variables ou opérandes.




| A | $\bar{A}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

| A | B | A ∧ B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

| A | B | A ∨ B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| A | B | A XOR B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Fig. 5.4 : Les tables de vérité des opérations logiques

## REPRESENTATION EXTERNE

L'information doit être représentée de façon externe d'une manière pratique. Habituellement les résultats sont obtenus sous forme de symboles alphanumériques et de nombres décimaux. Quand on veut avoir accès au contenu binaire proprement dit, on utilisera une représentation *hexadécimale.*

Cette représentation permet de représenter une suite de 4 bits (un quartet) à l'aide d'un seul symbole. Comme il faut 16 symboles ($2^4 = 16$) on utilise les 10 chiffres de 0 à 9 et les 6 premières lettres de l'alphabet (A à F).

De cette façon un octet peut être représenté à l'aide de deux chiffres hexadécimaux.

La table de conversion est donnée sur la fig. 5.5. A titre d'exemple, FF représente le nombre binaire « 1111 1111 ». La représentation hexadécimale est beaucoup plus facile à manipuler que la représentation binaire car elle est plus concise. De plus elle peut être facilement visualisée sur des afficheurs ou rentrée à l'aide d'un clavier à 16 touches. Sur les systèmes microprocesseurs les moins coûteux, toutes les entrée-sorties se font en hexadécimal.

**EXERCICE :** Quelle est la représentation hexadécimale de 10101010 ?
**REPONSE :** AA.

| DECIMAL | BINAIRE | HEXADECI-MAL | OCTAL |
|---|---|---|---|
| 0 | 0000 | 0 | 00 |
| 1 | 0001 | 1 | 01 |
| 2 | 0010 | 2 | 02 |
| 3 | 0011 | 3 | 03 |
| 4 | 0100 | 4 | 04 |
| 5 | 0101 | 5 | 05 |
| 6 | 0110 | 6 | 06 |
| 7 | 0111 | 7 | 07 |
| 8 | 1000 | 8 | 10 |
| 9 | 1001 | 9 | 11 |
| 10 | 1010 | A | 12 |
| 11 | 1011 | B | 13 |
| 12 | 1100 | C | 14 |
| 13 | 1101 | D | 15 |
| 14 | 1110 | E | 16 |
| 15 | 1111 | F | 17 |

Fig. 5.5 : Table de conversion en hexadécimal et en octal

## DEVELOPPEMENT D'UN PROGRAMME

La première étape est de tracer un organigramme (ou d'utiliser toute autre méthode équivalente). Un programme est alors écrit sur papier dans le langage choisi (assembleur ou langage de haut niveau). Le programme est entré à l'aide d'un clavier dans la mémoire du système (RAM). Il y aura toujours des erreurs de frappe durant cette phase, et c'est pour cela qu'il est souhaitable de disposer d'un moyen de faire des corrections, des insertions, des effacements.

Ces fonctions sont réalisées pour un programme dit *éditeur*. Un éditeur doit être disponible sur un systèoe pour faciliter l'entrée de données ou de programmes. Comme il n'est pas modifié par l'utilisateur, l'éditeur réside normalement sur une ROM. Sur certains systèmes, il pourra être stocké sur disque et amené en mémoire centrale avant son utilisation.

Après avoir été tapé et corrigé, le programme est mémorisé sous une forme symbolique. Il s'agit d'une séquence de caractères encodés dans le format ASCII décrit ci-dessus. En supposant que le

programme est écrit en langage assembleur, il doit être traduit en code-objet binaire directement exécutable. Pour cela, le programme *assembleur* sera chargé en mémoire centrale et son exécution produira une traduction du programme symbolique en code-objet. Cette traduction engendrera aussi des diagnostics pour l'utilisateur s'il y a des erreurs de syntaxe (l'assembleur détecte ces erreurs).

En supposant que l'assemblage se soit effectué sans erreur de syntaxe détectée, le code objet obtenu est alors prêt à être exécuté. Bien souvent cependant ce code objet a pu être stocké sur une mémoire auxiliaire et il est nécessaire d'utiliser un programme *chargeur* qui calcule les adresses et le positionne à l'endroit de la mémoire où il doit être exécuté. On suppose donc que maintenant ce code objet réside dans la mémoire RAM du système et est prêt à être exécuté. Bien peu de programmes s'exécutent correctement la première fois et si on lance un tel programme les chances qu'il y ait une erreur d'exécution sont grandes. Dans certains cas, il n'y a pas moyen de détecter ce qui n'a pas fonctionné, car rien de visible ne se passe !

Dans le but de déterminer ce qui ne va pas, un *programme d'aide à la mise au point* (« debugger » en anglais) est nécessaire. Il permet de suivre l'exécution et de localiser les endroits où il y a des erreurs.

Il permet de stopper l'exécution à des endroits spécifiés, d'examiner des registres ou des mémoires, de les modifier, etc. Avant d'exécuter un programme, l'utilisateur devra donc positionner des « points d'arrêt », en spécifiant des adresses auxquelles le programme d'aide à la mise au point arrêtera l'exécution du programme testé. Le programme s'exécutera alors sous contrôle du programme d'aide à la mise au point. Quand un point d'arrêt est atteint, l'utilisateur peut examiner les registres et la mémoire de manière à vérifier que l'exécution jusqu'à ce point a été correcte. Si cela n'est pas le cas, un segment de programme défectueux a été isolé. Le processus peut alors être raffiné en rajoutant d'autres points d'arrêts pour mieux localiser les erreurs. Habituellement plusieurs erreurs sont découvertes et l'on doit corriger ces erreurs avant de continuer. Quand le programme est exempt d'erreurs et s'exécute jusqu'au bout avec succès, il est présumé correct. (En effet dans beaucoup de programmes toutes les éventualités ne sont pas toujours testées et l'on peut découvrir des erreurs longtemps après la mise en place du programme). Le programmeur inexpérimenté pourra penser soit que son programme est parfaitement correct, soit au contraire que son programme ne sera jamais correct. La seconde solution est la

plus probable, car un programme ne peut pas se prémunir contre des situations qui n'ont pas été prévues par le programmeur.

Cependant ceci ne veut pas dire qu'un programme n'est jamais fiable. En pratique, le programme s'exécute normalement dans tous les cas usuels de façon indéfinie et ce n'est que dans certaines situations très particulières qu'une erreur ou un fonctionnement bizarre sera observé. Ces erreurs sont des erreurs quelquefois mineures mais qui sont difficiles à détecter. Ceci explique en particulier pourquoi les systèmes opératoires des gros ordinateurs sont si fragiles. Leur complexité est telle que les chances qu'une erreur encore indétectée soit déclenchée par une situation particulière, est non négligeable. Les programmes écrits sur microordinateurs n'ont pas cette complexité et l'assurance d'obtenir un programme correct dans un environnement donné est donc beaucoup plus grande. Ils sont donc aussi fiables que n'importe quel appareil mécanique utilisé quotidiennement (une voiture ?).

## LE LOGICIEL NECESSAIRE POUR LE DEVELOPPEMENT DES PROGRAMMES

On a vu que le logiciel minimum requis pour le développement de programmes consiste en : un éditeur, un assembleur, un interprêteur et/ou un compilateur pour les langages de haut niveau, et un programme d'aide à la mise au point. Tous les systèmes sérieux fournissent de telles facilités. De plus, si l'on utilise un disque, un système opératoire à partir du disque (DOS) sera nécessaire. Un DOS est un programme fourni par le constructeur et qui assure la gestion de la mémoire auxiliaire constituée par le ou les disques. La puissance d'un tel système varie d'un constructeur à l'autre. Dans certains cas il est possible de manipuler des fichiers symboliquement sans avoir à se préoccuper de leur emplacement physique. C'est le DOS qui se charge de toute la partie gestion de fichier. Cependant cela n'est pas toujours le cas sur les systèmes microordinateurs. Ceci sera discuté dans la section « disque » du chapitre périphériques.

Si un langage interactif tel que BASIC est utilisé, un interprêteur pour ce langage sera nécessaire. D'autre part certaines facilités additionnelles seront nécessaires pour des développements matériels

tels que simulation et émulation. Ces sujets ne seront pas abordés ici car ils dépassent l'objet de ce livre. (Voir réf. C4).

Finalement, on mentionnera le programme *moniteur*. Chaque microordinateur est livré avec un moniteur résidant dans une ROM. Ceci est nécessaire pour pouvoir décoder les commandes qui seront frappées au clavier. Chaque fois que le système démarre, une remise à zéro est effectuée et le microprocesseur commence à exécuter un programme qui réside à partir d'une adresse mémoire spécifique (la mémoire d'adresse 0 bien souvent). Ce programme remet à zéro les registres, vérifie que tout fonctionne normalement et se met en attente de commandes venant du clavier.

Lorsque l'utilisateur enfonce une touche du clavier, le programme moniteur lit le caractère correspondant et le traite pour déterminer s'il s'agit d'une commande.

Le moniteur inclut normalement un certain nombre de facilités d'aide à la mise au point telles que lecture ou écriture de registres et mémoires.

## AINSI VOUS VOULEZ PROGRAMMER ?

Vous avez deux possibilités :

Si vous voulez simplement utiliser un microordinateur, sans savoir ce qui se passe à l'intérieur, il n'y a pas de raison d'utiliser le langage assembleur qui manipule des registres. Dans ce cas vous utilisez un langage de haut niveau tel que le BASIC (voir chapitre suivant).

La deuxième alternative est de programmer en assembleur ou même en hexadécimal. Si vous êtes satisfait de l'hexadécimal ou avez un budget limité, tout ce dont vous avez besoin est une carte microordinateur avec une alimentation. La carte devra avoir un clavier pour entrer les instructions et les données, ainsi que des afficheurs à diodes luminescentes. Il en faut six en général pour pouvoir visualiser une adresse hexadécimale associée à la donnée en hexadécimal qui se trouve à cette adresse (2 digits pour la donnée, 4 pour l'adresse à 16 bits).

De plus, il sera souhaitable de disposer d'un interface avec un enregistreur à cassettes permettant de stocker les programmes.

Il faut noter cependant qu'un microordinateur personnel minimal, avec un clavier, un assembleur, un interprêteur BASIC et une unité de visualisation peut être acheté à un coût qui est sensiblement le double d'une carte microordinateur ! Dans le reste du livre on supposera donc que l'on dispose d'un tel outil.

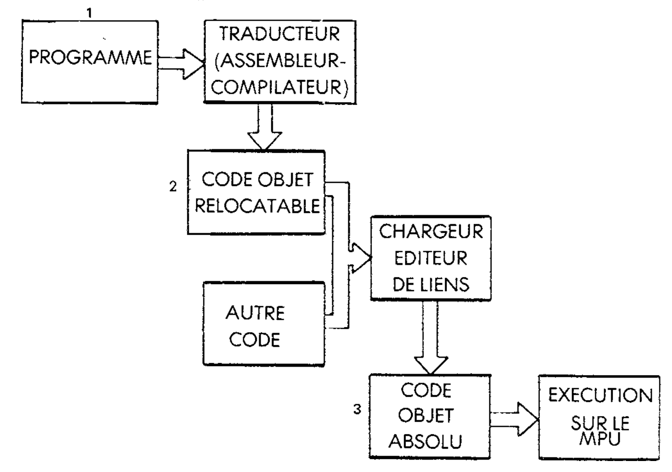



Fig. 5.6 : Une séquence de Développement de Programme




« Votre nouveau Programme, patron ! »

# CHAPITRE VI
# BASIC ET APL

BASIC est un langage de programmation de « haut niveau ». « BASIC » signifie « Beginners All purpose Symbolic Instruction Code » qui se traduit par « code d'instructions symbolique à usage général pour les débutants ». Ce langage a été développé au collège de Dartmouth.

Il s'agit essentiellement d'une version réduite du vénérable langage FORTRAN.

BASIC présente l'avantage par rapport au langage FORTRAN de pouvoir être appris facilement par des débutants.

De plus BASIC est un interprêteur c'est-à-dire que chaque instruction peut être exécutée immédiatement après avoir été frappée : c'est ce que l'on appelle un « interactif » qui permet le dialogue avec l'utilisateur. L'ordinateur détecte immédiatement les erreurs de syntaxe qui peuvent être corrigées tout de suite.

Nous allons d'abord examiner quelques exemples et observer les règles de programmation :

```
10 PRINT « LA SOMME DE 2 + 3 EST » 2 + 3
20 END
```

Chaque ligne est une instruction. L'exécution de ce programme est obtenue en tapant « RUN » et le résultat est l'impression (ou l'apparition sur l'écran) de :

```
LA SOMME DE 2 + 3 EST 5
```

Le texte entre guillemets a été imprimé tel quel. L'expression « 2 + 3 » a été « évaluée » à 5.

Chaque programme se termine par une instruction END (fin) et chaque instruction est précédée d'un numéro appelé étiquette (10 et 20 dans notre exemple). L'étiquette est nécessaire pour pouvoir classer les instructions. Si on veut ajouter une instruction, on peut taper par exemple :

```
9 PRINT « C'EST UNE ADDITION »
```

Cette instruction sera alors insérée automatiquement dans le programme précédent avant l'instruction 10 et on obtient le programme :

```
 9 PRINT « C'EST UNE ADDITION »
10 PRINT « LA SOMME DE 2 + 3 EST » 2 + 3
20 END
```

Ce programme imprimera alors « C'EST UNE ADDITION », puis sur la ligne suivante :

« LA SOMME DE 2 + 3 EST 5 »

En utilisant des étiquettes, il est possible d'ajouter de nouvelles instructions à n'importe quel moment. Pour cela, de manière traditionnelle, on étiquette initialement les instructions successives avec des dizaines, 10, 20, 30, etc. de façon à pouvoir insérer ultérieurement commodément de nouvelles étiquettes.

L'instruction 10 réalise ici une opération de type arithmétique. BASIC peut effectuer 5 types d'opérations :

+ — * / * * (ou ↑)

. + et — représentent l'addition et la soustraction

. * et / représentent la multiplication et la division

. ↑ ou * * indique l'élévation à la puissance, suivant les caractères disponibles sur l'imprimante et le clavier.

De plus, des parenthèses peuvent être utilisées pour spécifier des groupements de termes dans lesquels sont spécifiées les opérations à effectuer. Le programme précédent montre qu'il existe deux types d'instructions :

— les instructions exécutables telles que « PRINT »

— les commandes telles que END ou RUN.

Nous allons maintenant donner un exemple plus compliqué :

```
10 LET A = 2
20 LET B = 3
30 LET S = A + B
40 PRINT « LA SOMME DE A + B EST » S
50 END
```

Le résultat est le même que précédemment mais ici A, B, S sont appelées des *variables*. L'instruction « LET » assigne la valeur 2 à la variable A, 3 à la variable B et la valeur de la somme de A + B à S.

Les variables en BASIC « pur » se composent d'une seule lettre, suivie d'un chiffre optionnel.

Ceci peut être formalisé par :

[variable] = [lettre] ([chiffre])

où les parenthèses signifient « optionnel ». Exemples : A, A1, A5, B, B6, C4, F5. Ceci est une forme simplifiée de la notation BNF (« forme normalisée de Backus ») qui permet de représenter la syntaxe d'un langage.

Le mot LET est un « mot-clé » initialement utilisé dans le BASIC. Le BASIC « pur » continue à l'utiliser mais ce mot est en fait inutile et souvent peu pratique. Les BASIC les plus récents se dispensent souvent de l'utilisation explicite de « LET ». Les autres règles restent identiques. Ceci est naturellement un obstacle à la standardisation.

Comment sont représentés les nombres en BASIC ?

Le BASIC pur permet d'utiliser des nombres entiers et décimaux. Cependant, bien souvent, les BASIC fournis sur microordinateur ne fournissent que les nombres entiers. Ceci est alors un inconvénient important. De plus, il est nécessaire pour l'utilisateur de vérifier le nombre de chiffres permis par le BASIC fourni. 6 chiffres significatifs est un minimum, car cela ne permet pas de représenter de nombres plus grands que 9999.99 en décimal.

## LECTURE DE DONNEES

Deux instructions permettent de lire des valeurs associées à des données en BASIC. Ce sont :

« READ » (lire) et « INPUT » (entrée).

L'instruction READ utilisée en conjonction avec une instruction DATA (donnée) et permet d'assigner à une variable une valeur listée dans une instruction DATA. Exemple :

```
10 DATA 2
20 READ A
```

donnera A = 2

Pourquoi ne pas écrire directement LET A = 2 ? Parce que ce programme marchera ainsi avec divers « DATA » placés en tête de programme. Il pourra être commodément réutilisé avec des données différentes. Toutefois, le résultat est équivalent.

Des variables multiples peuvent aussi être assignées :

```
10 DATA 2,3,4,5,6
20 READ A,B,C,D,E
```

Le résultat sera A = 2, B = 3, C = 4, D = 5, E = 6. Ceci est pratique pour « entrer » des données multiples.

L'instruction « INPUT » permet la lecture directe d'une donnée à partir du clavier. Le programme s'arrêtera et un « ? » apparaîtra sur l'écran de visualisation ou l'imprimante. L'utilisateur tape alors un nombre, terminé par le caractère spécial « retour chariot » que nous avons déjà utilisé. La valeur sera alors lue par le programme qui continuera à exécuter. Exemple :

10 INPUT A
après exécution, l'ordinateur répond « ? »
l'utilisateur frappe 24 RC (retour chariot)
le programme continue avec la valeur A = 24.

**COMMENTAIRES**

Il est possible de clarifier un programme en y insérant des commentaires. Ceci est possible grâce à l'instruction « REM » (remarques).

```
10 REM CE PROGRAMME MULTIPLIE UN NOMBRE PAR
   2
20 INPUT A
30 PRINT « LE DOUBLE EST » 2 * A
40 GO TO 20
```

En réponse aux entrées successives 2,3,4, ce programme affichera 4,6,8. La quatrième instruction toutefois, est nouvelle. C'est une instruction «GO TO» (aller à). Elle signifie que au lieu d'exécuter l'instruction en séquence, le programme retournera à l'instruction 20 et réalisera une entrée de A à nouveau (l'instruction 20).

**QUESTION :** *Quand ce programme s'arrêtera-t-il ?*

**REPONSE :** *Jamais, de lui-même. Il faudra l'interrompre. Ceci est appelé une « boucle infinie ». Normalement il ne doit pas y avoir de boucle infinie dans un programme.*

**QUESTION :** *Que fait le programme suivant ?*

```
10 REM CALCUL DE LA TVA
20 INPUT P
30   REM L'UTILISATEUR DOIT FRAPPER LE PRIX
     D'ACHAT
40 T = P * 17,6/100
50 PRINT « LA TVA A 17,6 % EST » T
60 PRINT « SVP PAYEZ » (P + T)
70 PRINT
80 GO TO 20
```

**QUESTION :** *Que fait l'instruction 70 ?*

**REPONSE :** *Cette instruction insère une ligne blanche.*

## NOTATION SCIENTIFIQUE

Dans le but de ne pas gaspiller du temps et de la place sur papier, BASIC imprimera les nombres décimaux de façon normalisée sous forme de notation scientifique :

E + 05 représente $10^5$ (E représente l'exposant de la puissance de 10).

## INSTRUCTIONS DE TEST

La véritable puissance d'un programme ne vient pas tellement de la possibilité de faire des calculs mais de permettre de prendre des décisions en fonction des résultats d'instructions de test.

L'instruction :

```
10 IF A = 1 THEN 50
```

signifie que : « si A est égal à 1, alors il faut exécuter l'instruction 50 » (aller à l'instruction 50), sinon le programme continue en séquence. L'instruction IF ... THEN ... est puissante et peut être utilisée avec les opérateurs dits de relation qui sont :

= égal
< plus petit que (inférieur)
> plus grand que (supérieur)
> = supérieur ou égal
< = inférieur ou égal
< > différent de (n'est pas égal)

De plus une expression arithmétique peut suivre le IF. Après le THEN on peut avoir une commande.

## BOUCLES AUTOMATIQUES

Il est souvent nécessaire de répéter une opération jusqu'à ce qu'une condition soit satisfaite. Ceci peut être accompli par une instruction IF ... THEN ... en conjonction avec une instruction GO TO en fin de boucle.

Cependant lorsque une variable est régulièrement incrémentée, une facilité plus puissante existe : c'est la « boucle », réalisée avec l'instruction « FOR ... NEXT ». Exemple :

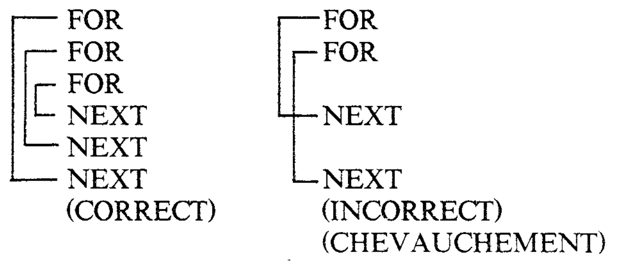
```
10 REM IMPRIMER LES NOMBRES DE 1 A 50
20 FOR A = 1 TO 50
30 PRINT A
40 NEXT A
50 END
```

Le programme assigne d'abord la valeur « 1 » à A et l'imprime (instructions 20 et 30). L'instruction 40 (« NEXT A ») a pour effet :

— d'incrémenter A de 1 : A devient égal à 2

— de forcer un « branchement à l'instruction FOR » c'est-à-dire l'instruction 20 ici. Le programme boucle jusqu'à ce que A atteigne 50. Quand A = 50, l'instruction 40 ne renverra pas à l'instruction 20 car A est égal à sa valeur limite 50 et le programme continuera en séquence. Ici le programme se termine (« END »).

De telles boucles peuvent être imbriquées les unes dans les autres à condition qu'elles ne se chevauchent pas :




Boucles imbriquées

## LES FONCTIONS INTERNES

Il est désirable de disposer de plus de facilités de calcul algébrique. Pour cela BASIC fournit des fonctions désignées par des sigles de 3 caractères et représentant des fonctions mathématiques usuelles :

INT ( ), SQR ( ) , ABS ( ) , SGN ( )

INT donne la valeur entière d'un nombre décimal : INT (—2.5) = 2

SQR est la racine carrée

ABS est la valeur absolue ABS (-2,5) = 2.5

SGN donne le signe : + 1 pour un nombre positif
- 1 pour un nombre négatif
0 pour 0

## SOUS-PROGRAMMES

Un sous-programme (« subroutine ») est un segment de programme écrit comme une unité séparée et qui peut être utilisé de façon répétitive par un programme principal. Exemple :

```
 (PROGRAMME PRINCIPAL)
 50  GOSUB 150
 60  •••
150 (SUBROUTINE)
200 RETURN
```

L'instruction 50 causera l'exécution d'un saut à l'instruction 150. Le sous-programme à l'adresse 150 sera exécuté jusqu'à ce qu'il rencontre l'instruction RETURN. Quand cette instruction est exécutée il y aura retour à l'instruction 60 qui suit l'appel du sous-programme (« subroutine »).

Cela revient à insérer de manière logique les instructions du sous-programme à l'endroit où il y a eu l'appel du sous-programme.

Ceci évite d'avoir à écrire explicitement le sous-programme à chaque fois que l'on en a besoin dans le programme principal. Il suffit pour cela d'utiliser l'instruction GO SUB. On peut ainsi créer à volonté des fonctions non fournies par le BASIC standard.

## FONCTIONS DEFINIES PAR L'UTILISATEUR

BASIC permet de plus à l'utilisateur de définir ses propres fonctions en utilisant l'instruction DEF FCN.

DEF FNC (X) = [expression]

L'avantage d'une fonction par rapport à une subroutine est qu'il suffit d'écrire son nom au sein d'une expression. Toutefois, elle est limitée à une expression.

## LES LISTES ET LES TABLEAUX

BASIC fournit deux structures de données : les listes unidimensionnelles et des tableaux à deux dimensions.

Le programme suivant lit 10 nombres au clavier et les imprime :

```
10 FOR I = 1 TO 10
20 INPUT L [I]
30 NEXT I
40 PRINT L [1] ; L [2] ; ... L [10]
```

L'instruction « FOR .... TO » spécifie une boucle : la variable I prend d'abord la valeur 1, et sera incrémentée par l'étape 30. Après l'étape 30 l'instruction suivante est l'étape 10 jusqu'à ce que I atteigne la valeur 10. Quand I atteint 10, c'est la fin de la boucle et l'instruction suivante 40 imprime la liste L qui aura été lue au clavier. L (1) est le premier élément de la liste, L (2) est le suivant, etc. Il s'agit d'un « tableau à une dimension ».

## EXEMPLE

Le programme suivant lit 10 nombres en provenance du clavier et calcule leur somme.

```
20 S = 0
25 REM LA SOMME S EST INITIALISEE A ZERO
30 FOR I = 1 TO 10
40 INPUT A [I]
50 S = S + A [I]
60 NEXT I
70 PRINT « LA SOMME EST » S
```

Le lecteur doit vérifier l'exactitude de ce programme en l'exécutant en séquence à la main (on dit « le faire tourner à la main »). S est la somme et prend d'abord la valeur 0, puis on lui ajoute la valeur de chacun des éléments A (I) lus à partir du clavier (étape 40) à l'intérieur de la boucle (étapes 30.40.50.60) et la somme totale est finalement imprimée (étape 70).

Ainsi, en 60, la première fois, S vaut 0 + A[1] = A[1]. La deuxième fois, S vaut A [1] + A [2], etc., et, à la fin, S = A [1] + A [2] + ... + A [10].

**EXEMPLE MODIFIE**

Le programme suivant lit puis imprime les 10 nombres puis le total T.

```
10 REM CALCUL DU TOTAL DES VENTES POUR 10 PRODUITS
20 FOR I = 1 TO 10
40 INPUT A [I]
50 NEXT I
60 PRINT « LES VENTES SONT AUJOURD'HUI : »
70 FOR J = 1 TO 10
80 PRINT A [J]
90 NEXT J
100 REM CALCUL DU TOTAL
110 T = 0
120 FOR K = 1 TO 10
130 T = T + S [K]
140 NEXT K
150 PRINT « LE TOTAL EST », T
```

**UNE TABULATION DE GESTION**

Soit une liste représentant le nombre d'unités vendues pour 10 produits à l'inventaire. L'instruction DATA indique les produits vendus pendant la journée, N (I) représente le nombre d'unités vendues pour chaque item I. On tape « 0 » pour finir la liste.

```
10 REM METTRE TOUS LES N A ZERO
20 FOR I = 1 TO 10
30 LET N [I] = 0
40 NEXT I
50 READ Q
60 REM TEST POUR SAVOIR SI L'ON A EPUISE LA LISTE
   DES PRODUITS VENDUS
70 IF Q = 0 THEN 100
80 LET N[Q] = N[Q] + 1
90 GO TO 50
100 PRINT « LES VENTES PAR PRODUIT SONT : »
```

```
110 FOR I = 1 TO 10
120 PRINT I, N [I]
200 DATA 5,2,1,1,1,,8,7,3,4,5,9,1,2,1,0
999 END
RUN

LES VENTES PAR PRODUIT SONT :
  1   5
  2   2
  3   1
  4   1
  5   1
  6   0
  7   1
  8   1
  9   1
 10   0
```

## FACILITES SPECIALES

BASIC fournit également des instructions pour manipuler des matrices, des chaînes de caractères, et des fichiers ainsi qu'une spécification de format de sortie (« TAB ») permettant d'effectuer des tabulations.

La plupart des BASIC permettent de définir plusieurs instructions sur une même ligne, séparées par des caractères , ; : ou / , ainsi que des affectations multiples telles que X,Y,Z = 0.

Dans certains cas, REM peut aussi être remplacé par une « * » ou par des commentaires entre apostrophes.

## VOTRE BASIC EST-IL COMPLET ?

Bien que BASIC soit en théorie, « standardisé », en pratique il n'y a pratiquement pas deux BASIC identiques. Les différences principales que l'on rencontre, c'est-à-dire les facilités qui peuvent le plus souvent manquer sont :

— le calcul sur nombre décimaux :

Cela peut être surprenant, mais la plupart des « mini-BASIC » fournissent seulement le calcul sur des nombres entiers !

— le nombre de chiffres significatifs pour représenter les nombres. Ce nombre devrait être au minimum de 6, mais 9 est

souvent nécessaire et plus de 9 est souhaitable pour les applications scientifiques ou commerciales.

— la disponibilité d'un système de fichiers de façon à pouvoir retrouver automatiquement des données et des programmes stockés sur disque ou sur un autre support.

— les fichiers séquentiels (bande magnétique ou cassettes) et les fichiers à accès aléatoires (disques).

— les instructions de formatage en sortie.

— les instructions de manipulation de chaînes de caractères.

De plus, l'éditeur fourni, doit être suffisamment puissant pour corriger les programmes : insertion, ajout, modification, recherche d'une chaîne de caractères, etc.

Des facilités supplémentaires peuvent améliorer les performances du BASIC :

— disponibilité d'un dispositif câblé de calcul en flottant (carte supplémentaire). Ceci peut accélérer la vitesse d'exécution des instructions arithmétiques par un facteur de 10 ou plus.

— l'appel à des programmes en langage machine : (CALL) de cette façon les séquences les plus utilisées d'un programme peuvent être codées en assembleur et converties en code objet (langage machine), appelées pour un programme BASIC, ce qui augmente la vitesse d'exécution de ces sous-programmes.

En résumé, les déficiences les plus courantes des BASIC sont l'absence d'instructions d'entrées-sorties avec formatage, de manipulation de chaînes de caractères ou de calcul arithmétique décimal avec une précision suffisante.

## LE BASIC DE GESTION

Le « BASIC gestion » est un BASIC complet équipé d'extensions permettant de traiter facilement du texte et de faire des éditions et des calculs en décimal.

Les extensions principales type, sont :

*1. Les entiers binaires (indiqués souvent par le symbole %).*

*2. L'arithmétique avec une précision étendue (12 à 18 chiffres significatifs).*

Il est important de se rappeler qu'un ordinateur tronque les nombres de façon interne dans les calculs et que de cette façon, des chiffres significatifs peuvent être perdus au cours d'opérations arithmétiques. Le problème de la propagation d'erreurs de calcul dépasse l'objet de ce livre. Cependant on peut par exemple avoir un

résultat de l'opération 3 * 1/3 qui ne soit pas un résultat égal à « 1 » mais à 0.999999999.

La plupart des calculs de gestion sont simples. Dans ce cas, l'erreur est minimale et très proche de zéro ou nulle. Pour obtenir une précision acceptable dans le résultat final, il est important de prendre en compte le maximum de chiffres. Pour les applications de gestion, 10 chiffres sont un minimum et 18 sont souhaitables.

*ATTENTION:*

Avant d'utiliser un BASIC pour des applications de gestion, il est fortement conseillé de réaliser les calculs arithmétiques les plus défavorables et d'observer la précision des résultats.

*3. Les formats de sortie.*

Dans les applications de gestion, il est essentiel de pouvoir engendrer des sorties imprimées de format approprié. La disponibilité d'instructions de ce type est donc importante. Le BASIC standard, comme le FORTRAN est connu pour sa difficulté et la lenteur de son traitement pour effectuer de telles mises en page.

*4. Chaînes de caractères blancs.*

Une caractéristique importante pour la mise en page est la possibilité de spécifier facilement des zones de blancs à l'impression.

*5. Recherche de chaînes de caractères.*

Quand on recherche dans une liste un code ou un nom, le programme doit rechercher une chaîne de caractères dans un ensemble de caractères. Les commandes qui permettent de réaliser une telle procédure facilement sont donc très utiles pour des applications de gestion. Il devient aussi beaucoup plus facile, de mettre à jour une liste, de l'ordonner alphabétiquement ou de rechercher des duplications.

*6. Assignation de chaînes de caractères.*

Quand on manipule des chaînes de caractères, il est intéressant de pouvoir les manipuler comme des données, c'est-à-dire de pouvoir leur donner un nom et les assigner à des variables de manière à pouvoir modifier leur contenu. Les modifications de contenu peuvent impliquer : des troncatures, des concaténations, des insertions, des substitutions.

*7. Accès direct au disque.*

De manière à pouvoir accéder efficacement (rapidement) aux

données sur le disque, des instructions spéciales doivent être fournies.

## QUELLE EST LA VITESSE D'EXECUTION DE VOTRE BASIC ?

L'efficacité d'un interprêteur BASIC est le facteur important. Naturellement, la vitesse du microprocesseur joue un rôle significatif. Cependant, un interprêteur mal réalisé peut être jusqu'à 100 fois plus lent qu'un autre réalisé de manière professionnelle. Ce n'est pas rare pour les microordinateurs.

Beaucoup de constructeurs sont amenés à réaliser rapidement des « mini-BASIC » qui fonctionnent correctement mais lentement et sont incomplets. Pendant ce temps ils développent une nouvelle version complète et plus rapide. Il est donc difficile d'évaluer la vitesse d'exécution d'un BASIC sans faire des tests.

Cela est fait en utilisant des programmes de « benchmarks ». Un programme de benchmark est un programme « typique » écrit par l'utilisateur pour une application donnée. En exécutant ce programme sur plusieurs ordinateurs on obtient des temps d'exécution sur lesquels on peut effectuer une comparaison précise. Cependant ce processus peut être difficile, car les instructions peuvent devoir être « légèrement » modifiées pour exécuter un même programme avec différents interprêteurs BASIC.

## LE BASIC COMPILE

La relative inefficacité au niveau de la vitesse d'exécution est intrinsèque à la nature des interprêteurs. *Un interprêteur* traduit chaque instruction en un format exécutable par la machine et l'exécute immédiatement. Une fois exécutée, il ne reste plus que le résultat de l'opération.

Cependant la plupart des programmes contiennent des *boucles* qui s'exécutent un grand nombre de fois. Dans une boucle, chaque instruction sera traduite puis exécutée n fois si la boucle est exécutée n fois.

Ceci est tout à fait inefficace du point de vue de la vitesse d'exécution, puisque les mêmes instructions sont retraduites chaque fois.

Un *compilateur,* traduira le programme utilisateur en une seule fois en un format directement exécutable (« code objet ») qui pourra

être exécuté à n'importe quel moment. Le code objet contient le code machine associé à une boucle sans avoir à répéter la traduction à chaque itération : l'exécution est donc beaucoup plus rapide, en principe (c'est-à-dire s'il s'agit d'un bon compilateur).

Cependant une fois que la traduction a été effectuée, il devient difficile d'ajouter, de modifier ou d'enlever des instructions et le programme doit être retraduit (compilé) dans son ensemble ce qui implique un délai plus grand que dans le cas d'un interprêteur. Pour cette raison, les langages compilés sont moins utilisés dans un environnement *interactif.* En fait BASIC fut développé comme une version interprêtée du langage FORTRAN (FORTRAN est un langage compilé).

Il existe maintenant des systèmes fournissant un BASIC compilé en plus de l'interprêteur. Le gain en vitesse d'exécution peut être important. (Plusieurs fois plus rapide, suivant le programme et l'efficacité du compilateur). Cette disponibilité est très utile pour les programmes opérationnels et terminés qui ne seront pas modifiés pour un certain temps. L'interprêteur par contre est très utile pour le développement de nouveaux programmes : il permet une mise au point rapide.

## LES LIMITATIONS DU BASIC

BASIC est, de loin, un des langages les plus faciles à utiliser mais il n'est pas complet :
1. ses possibilités de calculs mathématiques sont limitées par le nombre restreint des fonctions fournies, sa précision limitée et la lenteur d'exécution. De plus il ne peut traiter des tableaux à plus de deux dimensions (sauf en BASIC étendu).
2. ses possibilités de manipulation de caractères sont minimales (mais supérieures à FORTRAN) et il est quelquefois difficile de manipuler des structures de données complexes.
3. la syntaxe du langage ne permet pas d'avoir des structures de blocs ce qui limite la complexité théorique des programmes que l'on peut écrire dans ce langage.

## UN BASIC STANDARD ?

Deux BASIC sont désormais devenus standards :
1. Le BASIC de Microsoft, disponible pour presque tous les

microordinateurs, particulièrement efficace, et l'un des plus complets. Il s'agit d'un interprêteur.
2. Le CBASIC, ou BASIC commercial, un BASIC semi-compilé, fournissant un accès direct aux fichiers. Utilisé pour la majorité des applications de gestion, sous le système d'exploitation CP/M (voir la fin du chapitre 10 pour une description de CP/M).

## ALTERNATIVES A BASIC

D'un point de vue historique sur les microordinateurs la seule alternative à BASIC a longtemps été PL/M. (PL/M est une marque déposée d'INTEL). PL/M signifie « Programming Language for Microprocessors ». C'est un langage compilé dérivé du langage PL/I d'IBM ou plus exactement du dialecte XPL.

En tant que langage de haut niveau, PL/M offre les avantages de ces langages par rapport au langage machine : les programmes sont plus faciles et plus rapides à écrire. Il en a également les inconvénients : l'exécution est plus lente et les programmes occupent plus de place en mémoire.

PL/M a eu le mérite d'être le premier langage de haut niveau développé sur microprocesseurs. La plupart des constructeurs fournissent désormais des versions de PL/M sous des noms variés : MPL, XPLM, SMPL.

En pratique PL/M est utilisé avantageusement dans les applications industrielles. Cependant PL/M est relativement complexe, plus difficile à apprendre que BASIC, et ne possède que des instructions arithmétiques sur les entiers.

De ce fait, PL/M n'est pas utilisé pour les applications individuelles ou de gestion. La question demeure donc : « y a-t-il une alternative supérieure à BASIC ? ». On peut répondre : APL ou PASCAL. APL sera maintenant décrit brièvement.

## APL

APL est un langage inventé par Iverson et dont l'emploi s'étend lentement mais de façon régulière dans tous les types d'applications.

*APL est sans doute l'un des langages les mieux adaptés pour les applications individuelles et de gestion, à condition d'avoir une formation mathématique de base.*

Les raisons sont :

*1. APL s'apprend tout seul.*

Les règles de syntaxe d'APL sont si simples que l'on peut en quelques heures être capable de faire de petits programmes de calcul. APL est interactif et les erreurs de syntaxe sont détectées et signalées immédiatement (comme pour BASIC).

*2. APL est très puissant*

APL utilise des opérateurs spéciaux qui permettent de réaliser des opérations complexes sur n'importe quelle structure de données. Un seul opérateur permet par exemple de transposer une matrice à n dimensions ! De cette manière, des programmes relativement complexes peuvent être exprimés en quelques instructions. Ainsi APL est considéré comme un langage idéal pour les applications de gestion et est souvent enseigné dans les écoles supérieures de gestion.

*3. APL est un « langage lambda »*

On ne détaillera pas ici les principes théoriques du « calcul lambda », mais APL permet d'écrire des programmes aussi complexes que l'on veut, en définissant des variables ou des structures appropriées.

Naturellement APL possède des inconvénients :

*1. Il utilise un jeu de caractères spéciaux.*

Les opérateurs APL sont représentés par des symboles variés tels que ⍴ , ⍳ , ∇ , qui ne sont pas disponibles sur les claviers ou les imprimantes standards. Cependant ils le deviennent de plus en plus (en option).

*2. Les programmes APL sont si condensés qu'ils sont difficiles à déchiffrer.*

Cela résulte de la puissance du langage. Ce problème peut cependant être résolu en utilisant de bonnes habitudes de programmation et de documentation.

*3. Il est difficile de développer un interprêteur APL efficace.*

Ce problème a été résolu. Voir réf. Z10.

**Exemples simples**

— l'addition de deux vecteurs s'écrit :

1 2 3 4 5 + 2 3 4 5 6

et donne le résultat

3 5 7 9 11

— des opérations arithmétiques simples :

2 x 3

résultat imprimé aussitôt : 6

1 + 2 + 3 + 4
impression : 10
— impression d'un message :
« BONJOUR »
impression : « BONJOUR »
— un opérateur plus puissant :
! 6
calcule factorielle 6 c'est-à-dire 1 × 2 × 3 × 4 × 5 × 6 = 720
⌊ 2.3
L représente la valeur plancher de 2.3 c'est-à-dire 2
⍳ 5 (iota 5)
engendre le vecteur 1 2 3 4 5

**Un exemple plus complexe**

La factorielle de n est définie par :
n ! = 1 × 2 × 3 .... × n
Il s'agit du produit des n premiers nombres.

Le programme APL qui permet ce calcul (sans utiliser la fonction !) est :

```
∇ Z FAC N
[1] → 4 × ⍳ N = 0
[2] Z ← N × FAC N — 1
[3] → 0
[4] Z ← 1
[5] ∇
```

La première ligne du programme est la définition d'une fonction. Une fonction commence et se termine par le symbole ∇ .

Le nom de la fonction est FAC. Le paramètre est la variable N, le résultat est donc dans la variable Z. L'instruction [1] spécifie : aller à [4] si N = 0 (la combinaison × ⍳ peut être assimilée à un « SI ») l'instruction [2] spécifie : prendre la valeur N multiplié par FAC de (N — 1).

FAC N — 1 est un appel de fonction à l'intérieur de la définition de cette fonction. C'est ce que l'on appelle une fonction récursive. La fonction FAC est donc calculée à partir de FAC N — 1. L'instruction [3] correspond à aller à 0. C'est le retour de la fonction, c'est-à-dire, la fin du calcul de la fonction. A ce moment, Z doit contenir la valeur de la factorielle calculée.

[4] assigne une valeur 1 à Z.

La ligne [5] est la fin de la fonction (∇).

**Exercice**

*FAC 3 donne un résultat de Z = 6. Essayez de simuler l'exécution du programme à la main.*

## LA PROGRAMMATION EN LANGAGE NATUREL

Bien sûr le langage le plus pratique pour communiquer avec un ordinateur serait le langage naturel de l'utilisateur, le français par exemple. Il serait plus facile pour l'utilisateur de programmer directement en français.

Pendant les années d'expansion de la NASA et des programmes spatiaux, le gouvernement américain a financé un grand nombre de recherches pour déterminer la possibilité d'utiliser l'anglais ou un sous ensemble de cette langue pour programmer les ordinateurs. Ces recherches ont eu pour résultat de montrer que le langage habituel ne pouvait pas être utilisé.

Le problème en est son ambigüité. Le langage parlé n'est pas assez précis pour être sans équivoque et il faut tenir compte du contexte pour obtenir le sens d'une phrase. Le contexte peut être sensoriel (vision, geste, odeur) ou syntaxique et dépendre de ce qui a été déjà dit ou le sera. Le traitement nécessaire pour résoudre certaines ambigüités pourrait demander des calculs très longs sans assurance de succès, car dans certains cas la machine ne peut pas résoudre les ambigüités, notamment si le contexte est sensoriel ou culturel. Une telle approche est donc tout à fait inefficace.

Il faut utiliser un sous-ensemble du langage naturel ayant une structure propre et sans ambigüité. C'est ce que l'on appelle un langage de programmation. Une grande variété de tels langages ont été développés qui répondent aux besoins de différentes classes d'utilisateurs. Le plus simple à apprendre semble être BASIC. C'est ce qui explique le succès de BASIC dans le monde des microordinateurs.

On peut donc résumer : il n'y a pas d'espoir de voir un ordinateur comprendre directement un langage naturel dans le futur proche. Toute publicité indiquant une possibilité de programmation en langage « naturel » est donc inexacte. Il peut cependant exister des langages qui sont proches du langage naturel, mais qui n'en utilisent qu'un faible sous-ensemble, et doivent respecter des règles de syntaxe précises et sans ambigüités.

Il n'est pas interdit de « parler à l'ordinateur » dans un sous-ensemble du langage naturel qui soit facilement compris par la machine et ses utilisateurs. Dans ce cas on a simplement défini un autre langage de programmation de haut-niveau !

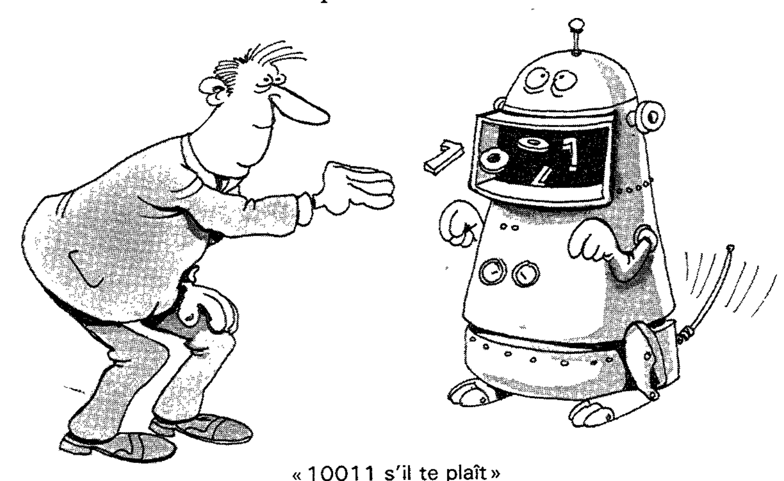
## RECAPITULATION DES INSTRUCTIONS BASIC

| | |
|---|---|
| CHANGE A TO A | Stocke le code ASCII de caractères dans un tableau A, avec la longueur de la chaîne de caractères. |
| DATA | Utilisé avec un ordre de lecture (READ) pour stocker des valeurs dans des variables séparées par des virgules. |
| DEF FNX (Y) | Définition de fonction. X est le nom de la fonction, Y est l'argument. |
| DIM A(X) | Dimension d'un tableau A (peut être d'1 ou 2 dimensions). |
| GO SUB N | Appel à sous-programme. Le sous-programme est exécuté jusqu'à l'instruction RETURN. |
| END | Dernière instruction d'un programme. |
| FOR I = A TO B STEP C | Instruction de boucle. L'incrément (STEP) est optionnel. |
| GO TO n | Spécifie l'instruction n comme la prochaine instruction à exécuter. |
| IF X THEN Y | Le test logique X est effectué. S'il est positif, l'instruction y est exécutée. Sinon l'instruction en séquence est exécutée. |
| INPUT | Permet de lire des données à partir du clavier jusqu'à un retour chariot (CR). |
| LET | Instruction d'assignation. Dans certains cas, le mot LET peut être omis et des assignations multiples peuvent être réalisées sur la même ligne. |
| MAT | Instructions sur matrices. |
| PRINT | Imprime des valeurs ou du texte séparés par virgules ou point-virgules. |

| | |
|---|---|
| NEXT I | Fin d'une boucle. Incrémente I et retour automatique à l'instruction suivant le FOR... |
| READ | Utilisé avec l'ordre DATA pour assigner une valeur à des variables. |
| REM | Remarque ou commentaire inséré dans un programme (instruction non exécutable). |
| RESTORE | Remet les instructions DATA au début de la liste. |
| RETURN | Dernière instruction exécutée dans un sous-programme. Permet le retour à l'instruction suivant un GO SUB. |
| STOP | Arrête l'exécution d'un programme. |

**AUTRES INSTRUCTIONS**

| | |
|---|---|
| FONCTIONS | Fonctions mathématiques prédéfinies en BASIC, telles que ABS (X), RND (X), SGN (X), SQR (X), SIN (X), COS (X), TAN (X), LOG (X), EXP (X), TAB (X). |
| FILE S | Instructions de manipulation de fichier : identification, lecture, écriture, insertion, ajout, etc. Les fichiers peuvent être en ASCII ou binaire, formatés ou non, séquentiels ou à accès aléatoire. |



« 10011 s'il te plaît »

# CHAPITRE VII
# APPLICATIONS DE GESTION

## INTRODUCTION

Pour la première fois le progrès de la technologie rend possible l'utilisation des possibilités d'un ordinateur dans l'environnement commercial de petites entreprises pour des coûts inférieurs à 50 000 F.

*Ceci est-il tout-à-fait vrai ?* La réponse est « oui, mais... »

Le but de ce chapitre est de justifier le « oui » et d'expliquer le « mais ».

*Un système microordinateur peut-il assurer le traitement de données commerciales ? Oui.*

*Y a t-il actuellement des systèmes qui le font ?* Non. (Toutefois, ils commencent à apparaître).

Nous verrons que la lacune essentielle des systèmes microordinateurs n'est pas le matériel mais le logiciel. Cela a toujours été le cas, depuis que les ordinateurs existent, et à chaque nouvelle génération de matériel le problème se retrouve. Nous verrons que le matériel nécessaire et suffisant pour traiter un grand nombre d'applications de gestion peut être acheté à des prix allant de 5 000 à 20 000 F. Cependant le logiciel commence seulement maintenant à être disponible. Bien sûr, il existe de nombreux compromis pour chaque système en fonction des services que l'on attend et des ressources disponibles.

Nous étudierons donc les applications classiques de gestion dans le but de définir les capacités de traitement requises pour réaliser ces objectifs.

Dans le but de permettre au responsable de faire un choix raisonnable, *il est impératif qu'il connaisse et comprenne* ces compromis pour évaluer les différentes solutions disponibles car il n'y a pas de *« solution la meilleure »* dans tous les cas. Le choix peut être comparé à la sélection d'une voiture ou d'une machine destinée à une application spécifique. Il n'y a pas de choix unique qui convienne à toutes les applications.

La compréhension et la connaissance du *matériel* (« hardware ») sont chose relativement simple. Le problème le plus complexe et le

plus difficile est la compréhension du *logiciel* nécessaire. Un grand nombre de responsables achèteront un système de gestion en commettant des erreurs. Les erreurs concernant un mauvais choix de logiciel sont généralement plus graves et plus coûteuses que celles concernant le matériel.

L'investissement en *logiciel* devient de plus en plus le coût dominant. Un système inadéquat empêchera la croissance des possibilités de système et donc la croissance de l'entreprise ou du service sans changement de système. La transition d'un système à un autre est toujours coûteuse et nécessite de nombreuses réorganisations. Pour ces raisons, le lecteur est fortement encouragé à étudier et à comprendre les concepts logiciels et matériels qui vont être présentés.

## APPLICATIONS DES ORDINATEURS A LA GESTION

Chaque entreprise doit tenir à jour un certain nombre de fichiers. Un « fichier » est simplement un bloc muni d'un nom. Les fichiers les plus courants sont : les comptes créditeurs et débiteurs, les inventaires, la comptabilité générale. D'autres fichiers sont généralement souhaitables, tels que : le fichier du personnel, la liste des clients, des fichiers d'adresses, des fichiers de ventes, de commandes, de fournisseurs, la trésorerie, les investissements mobiliers, les comptes de banque, et autres.

Ces listes sont actuellement traitées soit de façon manuelle (ce qui implique un personnel administratif), soit à l'aide d'appareils électromécaniques ou enfin à l'aide d'ordinateurs, et, le plus souvent, par une combinaison des méthodes précédentes.

En plus de la maintenance de ces fichiers, toute affaire applique des techniques de *traitement* spécifiques à chacun de ces fichiers. Par exemple, un programme de *paye* traitera le *fichier du personnel* et engendrera des bulletins de salaire et des ordres de versements ou des chèques. D'autres programmes traiteront des fichiers de vente pour produire des rapports de TVA à payer à l'Etat.

Un programme de gestion de procédure de transaction permettra de mettre à jour des fichiers spécifiques, de changer, de mettre à jour ou d'entrer de nouvelles données. Un exemple type est le cas d'une nouvelle vente : un programme transactionnel utilisera le fichier inventaire, le fichier fournisseurs, le fichier clients et peut-être d'autres. Il les mettra à jour et établira un certain nombre de rapports : factures, commandes, etc.

De façon similaire, une procédure d'arrivée de produits se chargera de gérer les arrivages, mettant à jour l'inventaire, la comptabilité, les fournisseurs, et la liste des commandes en attente.

Chaque *paiement reçu* nécessite la mise à jour de la *liste des comptes clients* et de la *trésorerie.*

Ces listes et les programmes type associés sont illustrés sur la figure 7.1.

Les flèches indiquent l'effet d'une transaction spécifique sur les différents fichiers du système.

En plus des programmes principaux qui sont indiqués par des cercles sur la figure, un certain nombre de journaux sont nécessaires pour produire les rapports nécessaires à la gestion commerciale et administrative de l'entreprise. Ceci sera décrit avec plus de détails dans les pages suivantes. Il est important de noter que les principes de base sont tout à fait simples :

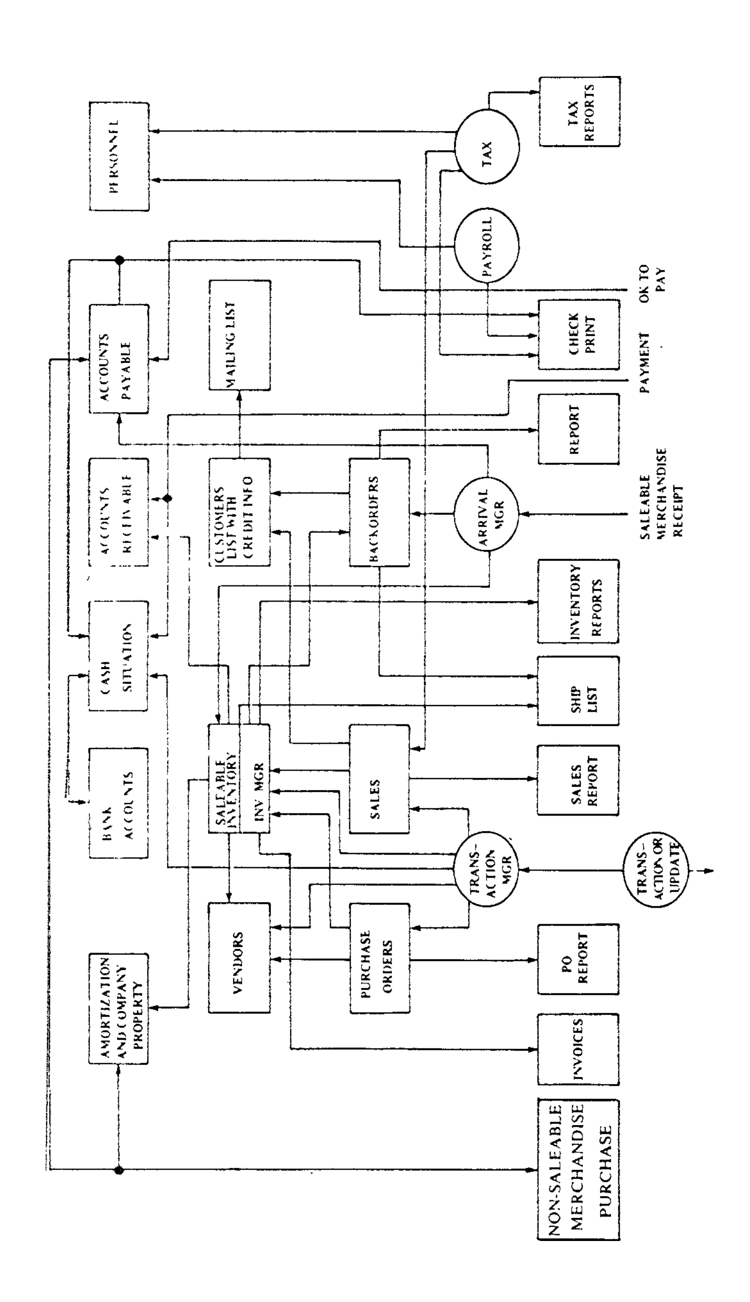
1 - Les fichiers doivent être créés et entretenus,

2 - Les programmes permettront de réaliser l'interface entre l'utilisateur et les fichiers et fourniront les fonctions de traitement requises.

Malheureusement dans une situation concrète de système de gestion, ceci est simplement *une partie* du traitement nécessaire. En fait, la maintenance d'un seul fichier est relativement simple dans la plupart des cas. Le plus gros du traitement nécessaire est dû à toutes les références croisées qu'il faut effectuer simultanément et aux mises à jour automatiques de fichiers multiples.

Prenons un exemple : une commande est reçue par courrier. Elle sera traitée par le programme de *gestion des transactions* sur la gauche de l'illustration. La vente devra être entrée dans le *journal des ventes* de la journée. Une séquence complexe d'évènements va se dérouler comme résultat de cette entrée. Le nom du client sera ajouté au fichier *clients* automatiquement. De plus, en fonction de la marchandise, de la quantité, de la profession du client, des codes associés à ce client seront répertoriés. De plus, il est probable que des informations sur la solvabilité du client seront recherchées par le *programme de solvabilité* avant que la commande soit acceptée. Ensuite, si la commande est acceptée par le programme de *gestion de commandes,* les étapes suivantes consisteront à honorer cette commande.

Pour cela l'*inventaire des marchandises* ou produits en stock devra être consulté pour voir si ces produits sont disponibles. Dans




APPLICATIONS DE GESTION

notre exemple supposons que trois produits A, B, C soient commandés. Seuls A et B sont en stock, mais pas C.

Une *facture* est alors engendrée (en bas à gauche de la figure 7.1), une *liste de produits à expédier* (en bas au milieu de la figure 7.1) et une *commande de réapprovisionnement* sont engendrées.

Dans notre exemple le produit B est en stock mais il ne reste plus que quatre unités de ce produit.

Le fichier inventaire inclut un champ spécial qui spécifie le *niveau de réapprovisionnement*. Le niveau de réapprovisionnement de B est quatre. Aussi un des résultats de cette transaction sera d'engendrer une *demande de réapprovisionnement* du produit B pour une quantité standard de 25 unités (spécifiée dans le fichier inventaire).

L'adresse du fournisseur est obtenue à partir du fichier fournisseurs en utilisant un numéro qui sert d'index dans ce fichier (gauche de la fig. 7.1).

*Cette simple transaction commerciale a nécessité l'utilisation de cinq fichiers et de plusieurs programmes de traitement.* Dans certains cas il pourrait y avoir encore d'autres fichiers à mettre à jour ou à vérifier.

Cet exemple montre donc clairement que, pour être utile, un système de gestion doit fournir des moyens d'accès faciles et des méthodes de mise à jour simultanées pour un grand nombre de fichiers. De plus, tout cela doit s'enchaîner de façon automatique sans intervention manuelle. Malheureusement, on verra que *la majorité des systèmes de gestion disponibles aujourd'hui et utilisant des microordinateurs ne proposent pas encore tous ces services.* Ils fournissent des manipulations de *fichiers simples* sans automatiser le processus transactionnel complet. Il y a donc encore beaucoup de choses à « faire à la main ».

## LE TRAITEMENT DE TEXTE

Le terme de « traitement de texte » fait référence à une opération de frappe à la machine automatisée, où l'utilisateur peut changer,

modifier ou mettre en page un texte. Il nécessite un programme éditeur de texte qui est disponible sur la plupart des ordinateurs. Le coût des microprocesseurs est devenu tellement bas qu'ils peuvent être dédiés à une seule fonction. En fait, le traitement autonome de texte devient de plus en plus disponible sur les machines à écrire électroniques.

Au contraire, les systèmes de gestion offrent des systèmes multiterminaux utilisant un tube cathodique similaire à un écran de télévision.

## UTILISATION D'UN SYSTEME AUTOMATISE DE GESTION

Nous allons maintenant utiliser un microordinateur pour une transaction commerciale simple. Nous spécifierons le type de programme, et nos choix en réponses aux questions apparaîtront sur l'écran d'un terminal.

Au début, le système visualise un « menu ». Un « menu » est simplement une question à choix multiples.

La question posée par le système est terminée par un ou plusieurs caractères d'attente de réponse (ici « ... ») indiquant que le microcalculateur est en attente d'une réponse.

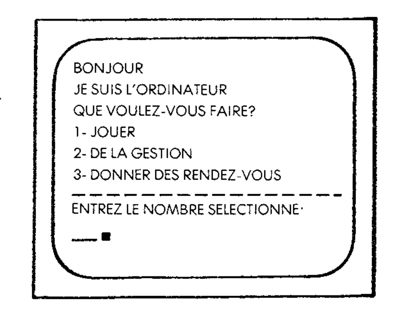



Fig. 7.2 : Un menu

Nous tapons « 2 ». Le programme de gestion est alors sélectionné. Le système (DOS) le charge automatiquement à partir du disque. Une liste apparaît alors à nouveau.

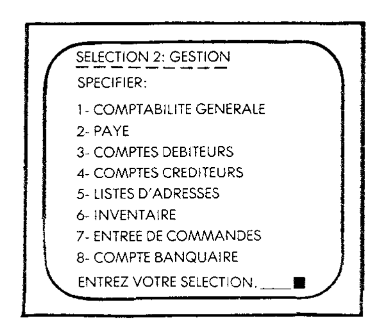



Fig. 7.3 : Le « sous-menu » GESTION

Nous choisissons : « COMPTE CREDITEURS ». A ce point, le système pourrait demander à ce qu'une nouvelle diskette soit introduite. Supposons que non et continuons.

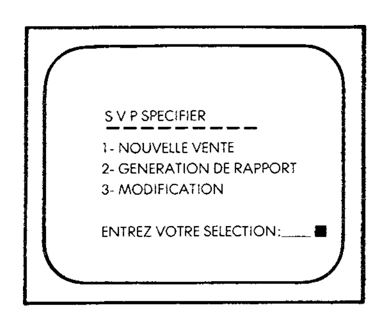



Fig. 7.4 : Le menu « Comptes créditeurs »

Si nous spécifions « nouvelle vente », le système demandera toutes les données concernant cette transaction, engendrera une facture puis mettra à jour le journal de comptes créditeurs, l'inventaire, et la liste clients. Le dialogue devient alors interactif et

le système demande toutes les informations n qualification de la vente.

```
INDIQUER LA DATE:. . . . . 060978 OK
              FACTURE
CLIENT:         COMPAGNIE 2000
                10 RUE DE L'AVENIR
                75012 PARIS
EXPEDIER A:     MEME ADRESSE
No DE FACTURE:          B1025 OK
DATE:                   060978 OK
No BON COMMANDE:  1984
PRODUITS VENDUS
REFERENCE    QTE    REMISE
D2001         5       0
F75125        6       0
FIN
```

Fig. 7.5 : Entrée d'une commande

```
LES PRIX SONT:
D2001     5  A  20,00. . . . . . . 100,00 F
D75125    6  A  30,00. . . . . . . 180,00 F
   TOTAL HT                      280,00 F
COUT D'EXPEDITION. . . . . . . . 22,00 F
EXEMPTE TVA . . . . . . . . . . . . . . NON
TAUX TVA....6%. . . . . . . . . .  16,80 F
TOTAL TTC. . . . . . . . . . . . . 318,80 F
DELAI DE PAIEMENT. 30 J ?. . . . . . OUI
```

Fig. 7.6 : Entrée d'une commande (suite)

```
VENDEUR. . . . . . . . . . . . . . . . . . . . D.R.
COMMENTAIRES SPECIAUX SUR
FACTURE . . . . . . . . . . . . . . . NON
COMMISSION VENDEUR. . . . . . NON
EXPEDITION. . . . . . . . . . . . . . . . . PTT
________________________________________
TRANSACTION TERMINEE          OUI
TRANSACTION SUIVANTE          NON
```

Fig. 7.7 : Entrée d'une commande (fin)

La transaction est maintenant terminée.

Le mode d'interaction avec le système devrait maintenant être clair. Le programme pose toutes les questions nécessaires, imposant ainsi une discipline pour l'entrée des données.

En même temps, le programme peut aussi tester la validité des données entrées (il détectera des valeurs illogiques). Finalement il imprimera automatiquement une facture et mettra à jour les fichiers et journaux nécessaires.

Examinons maintenant les caractéristiques d'un tel système.

## CARACTERISTIQUES ET CONTRAINTES D'UN SYSTEME DE GESTION

Les caractéristiques vont être analysées en fonction des fichiers essentiels qui doivent être gérés en fonction des principaux types de traitement effectué sur ces fichiers.

### Comptes à recevoir

Il s'agit du fichier (ou « journal » en comptabilité) qui contient une copie de toutes les factures engendrées par le système. Bien sûr, ce fichier ne contient pas une copie véritable de ces factures, mais simplement l'information qui est nécessaire pour engendrer de telles factures.

De manière typique, il y aura la date de la transaction, le nom et l'adresse du client, l'endroit où se fera l'expédition et le détail des produits vendus.

Il n'est pas nécessaire de mémoriser dans ce fichier toutes les informations qui apparaissent sur une facture. En effet, s'il existe un journal des ventes, toutes ces informations supplémentaires seront contenues dans ce fichier auquel on doit pouvoir accéder facilement et rapidement.

Le traitement efficace de l'information par ordinateur exige que tous les éléments d'un fichier soient de *longueur fixe*. Pour cette raison, les fichiers utilisent le plus souvent des blocs de longueur fixe appelés enregistrements ou « blocs ». Un fichier de comptes créditeurs peut être structuré de cette manière.

Des champs de longueur fixe peuvent être ainsi alloués à la date, au nom, à l'adresse, à la somme dûe, au code client, au n° de facture.

La présence d'un n° de facture permet d'accéder au reste de l'information dans le journal des ventes ou fichier factures.

Dans le jargon informatique, la présence de ce numéro est un *« pointeur »* vers des informations stockées ailleurs. Dans le jargon de la gestion ce numéro permet de retrouver le chemin de toute transaction. Le *journal* des comptes clients doit être distingué du *programme* de traitement « *audit trail* » associé. Le *journal* contient simplement une liste de tous les comptes. Les avantages et inconvénients de son format sont faciles à évaluer par l'utilisateur en fonction de ses besoins.

Les spécifications devront préciser tous les champs auxquels l'utilisateur désire accéder facilement et fréquemment.

Le *programme* de traitement, quant à lui, est responsable des manipulations de fichiers, de sa mise à jour et de la génération de différents états de sortie. Il doit pouvoir engendrer par exemple des listes des comptes anciens de plus de 30, 45, 60 ou 90 jours. Le programme est d'ailleurs habituellement conçu pour engendrer automatiquement des rappels aux clients.

Cependant un tel programme peut aussi être distinct du précédent. Dans ce cas, le programme de gestion des comptes créditeurs est utilisé pour engendrer un fichier des comptes clients ayant dépassé la date de paiement attendue. Ce fichier est alors utilisé par un programme spécifique engendrant des notices de rappel à tous les clients débiteurs ayant dépassé la date de paiement prévue.

Le fait de séparer des fonctions en programmes individuels ou d'intégrer toutes les fonctions dans un seul programme a peu d'impact sur la valeur du système. Il s'agit bien souvent d'un choix de programmation fait par le concepteur.

Le point important est simplement que *toutes les facilités désirées soient disponibles.*

### Les comptes débiteurs

Le fichier des comptes débiteurs est une liste de factures *reçues* par l'entreprise. Habituellement, quand un ordre de paiement a été validé par un responsable, le programme imprimera automatiquement les chèques de paiement pour les marchandises ou services reçus. Le chèque sera imprimé soit à une date spécifique, ou à une date programmée telle que trente jours après la réception de la facture. (Un bon programme d'impression de chèques doit aussi vérifier que le solde du compte bancaire est suffisant pour couvrir le montant des dépenses engagées !).

**L'inventaire**

Il n'y a pas de fichier inventaire standard, ils diffèrent d'une entreprise à une autre et bien souvent d'un produit à l'autre. Pour cette raison, les fichiers inventaires généraux prennent en compte un grand nombre de catégories qui ne seront pas nécessairement utilisées par une entreprise. L'absence de certaines pourrait être considérée comme une faiblesse du système. D'un autre côté, la disponibilité de trop de catégories signifie beaucoup de place perdue en mémoire dans le système. Cela se traduit par un nombre relativement plus faible de produits pris en compte dans l'inventaire. Cependant avec le coût décroissant des mémoires, le compromis est maintenant de fournir autant de catégories que possible pour la plupart des entreprises, même si certaines ne sont pas utilisées.

Rappelons que la taille du fichier inventaire est limitée par la place disponible sur un support de mémoire secondaire type tel qu'une disquette par exemple.

Les informations type qui peuvent être inclues dans un fichier inventaire sont : CODE - NO ITEM - DESCRIPTION PRODUIT - ENDROIT DU STOCKAGE - NOMBRE DISPONIBLE - NO FOURNISSEUR - PRIX D'ACHAT - COUT DE STOCKAGE - DATE DERNIERE VENTE - QUANTITE MINIMUM POUR REAPPROVISIONNEMENT.

De 64 à 128 octets seront nécessaires pour stocker ainsi un produit. En utilisant un tel format, on peut stocker de 1 800 à 3 600 items sur une disquette normale.

Le programme de gestion de stocks, quant à lui, doit fournir des fonctions multiples :

- Maintenance complète de l'inventaire, incluant les mises à jour automatiques de certaines informations
- Prise en compte des commandes
- Prise en compte des réapprovisionnements
- Histoire des ventes
- Commandes en attente
- Liste par qualité, classe, coût, fournisseurs, no produit, date de vente.
- Recherche des quantités minimum
- Mise à jour sélective
- Rapports d'activité
- Listes d'inventaires en fonction d'une combinaison de critères.

A titre indicatif, un système simple d'inventaire, écrit en BASIC nécessitera peut-être 10 K mots de mémoire (dans le cas des microprocesseurs un mot est égal à 1 octet). Un programme plus général nécessitera 90 K ou plus. Comme la mémoire d'un microordinateur ne peut dépasser 64 K octets, des techniques de recouvrement doivent être utilisées. Elles consistent à exécuter une partie du programme puis à recouvrir en mémoire une partie du programme déjà exécutée par une nouvelle partie du programme. Le programme complet n'est de ce fait jamais complètement chargé dans la mémoire. Les morceaux nécessaires sont amenés en mémoire au fur et à mesure des besoins. Naturellement ceci réduit l'efficacité du traitement du point de vue de la vitesse d'exécution.

Cependant, si les recouvrements sont écrits intelligemment, l'impact sur la vitesse d'exécution peut être modéré.

## Mise à jour

Il est important de noter encore une fois, que, techniquement, une mise à jour d'un fichier inventaire peut être réalisée « à la main ». L'utilisateur peut examiner une liste d'items dans un inventaire et modifier n'importe quelle entrée telle que, disons, le coût inventaire. Cependant la valeur d'un système informatique est de permettre la mise à jour *automatique* de toutes les informations interdépendantes dans plusieurs fichiers.

Ainsi un système de gestion complet doit pouvoir mettre à jour automatiquement le fichier inventaire, à chaque fois qu'une information liée au stock est changée quelque part. Par exemple, si le coût unitaire d'un produit est modifié, ce coût devrait être mis à jour automatiquement dans le fichier inventaire ainsi que dans tout autre fichier où il doit y avoir modification.

## Le fichier d'adresses

Le fichier d'adresses, en tant que fichier spécialisé, est souvent négligé dans un système de gestion.

En effet la plupart des systèmes de gestion fournissent un rapport complet des ventes. Dans ce fichier ventes, le nom des clients et toutes les informations associées à la transaction ont été enregistrées. A première vue, il semble que le système dispose de toute l'information pour lister les clients en vue d'une action de promotion par exemple. Techniquement cela est correct, mais en pratique une

telle liste n'est pas utilisable comme fichier d'adresses, du moins pour une utilisation immédiate et efficace.

Pour des envois postaux peu importants et de petits fichiers, la liste des ventes peut être traitée par un programme spécialisé qui sélectionnera les noms et les adresses et les imprimera.

Toutefois, pour des envois postaux importants (plusieurs milliers de noms), la liste des ventes est pratiquement inutilisable. Tout d'abord, le traitement serait trop long car il faut rechercher l'information au milieu d'un ensemble de données inutiles. D'autre part, si cette liste est contenue sur plusieurs diskettes cela est impraticable. Enfin, le traitement nécessaire serait complexe : élimination des noms en double, classement par code postal, sélections diverses.

Pour avoir un fichier d'adresses efficace, il est impératif que l'utilisateur code chaque transaction au moment où elle est réalisée. Par exemple, l'utilisateur codera le type de l'entreprise ou de la profession du client associé à la transaction. Ce code peut alors être utilisé pour des sélections sur critères dans le fichier d'adresses. Un fichier d'adresses peut être obtenu en traitant le fichier vente ou celui des clients de façon régulière, mais il doit en être distinct.

Le fichier doit être continuellement classé soit par ordre alphabétique soit par code postal ou selon tout autre critère. De cette façon, chaque fois qu'un nouveau nom est entré, on peut rechercher s'il y a duplication d'adresses. Même dans ce cas, une liste d'adresses contient un grand nombre de caractères pour chaque adresse. Elle contient : le nom, le prénom, la profession, le nom de la société, la rue et le numéro ou la boîte postale, la ville, le code postal et le pays. De plus, chaque adresse contiendra un code de 5 à 20 digits permettant d'effectuer les sélections sur critères.

Supposons que 128 caractères soient suffisants. Un secteur de disquette contient 256 octets, il est donc possible de stocker 2 adresses par secteur. Sur une disquette contenant 1 200 secteurs, on pourra donc enregistrer jusqu'à 2 400 adresses.

*Cela n'est pas suffisant pour la plupart des listes d'adresses.* De tels fichiers doivent pouvoir contenir jusqu'à 10 000 ou 50 000 noms.

Un tel système sera difficile à utiliser si plusieurs disquettes doivent être manipulées. Cela peut être inacceptable dans certains cas. Nous verrons que ceci requiert un « disque dur » de grande capacité.

Du fait de cette contrainte des microordinateurs, à savoir la

quantité limitée de mémoire à disque, il faut utiliser un fichier structuré de manière efficace. Un index est par exemple utilisé avec quelques octets par entrée. Il contient le nom ou code client et est associé à un pointeur. De cette façon, chaque entrée peut être décrite par 20 ou 30 caractères. Une disquette complète peut alors contenir un index correspondant au fichier complet, permettant un accès rapide.

Chaque fois qu'une sélection doit être faite, la disquette contenant l'index doit être chargée d'abord. Ainsi par exemple, nous voulons imprimer la liste de toutes les sociétés qui ont commandé pour plus de 1 000 F de pièces pendant les trois derniers mois. Le programme de sélection engendrera une liste de numéros de clients répondant à ces critères à partir du fichier index. Cette liste sera ensuite utilisée par un programme d'impression qui imprimera les adresses des clients à partir du fichier clients ou de celui des ventes.

Chaque fois qu'une sélection aura été réalisée, le programme commence à engendrer des messages tels que :

« SVP charger la disquette n° 2-5 »

Cette disquette contient le premier bloc de données que l'on doit imprimer. Un certain nombre d'autres disquettes devront être chargées, si la liste est longue. Cependant la sélection complète, qui est le processus de traitement le plus long, a pu être réalisée en un temps très court en utilisant une disquette index. C'est la limitation de capacité d'une disquette qui crée ici un inconvénient en requérant des chargements successifs.

Un bon système signalera sur un écran de visualisation les disquettes qui doivent être montées. L'utilisateur aura le temps de monter la disquette suivante pendant que le système imprime les données relatives à la disquette précédente. Au moment où on aura fini d'écrire par exemple la dernière étiquette de la disquette précédente, la disquette suivante aura déjà été montée.

Un tel système fonctionnera donc à la vitesse maximale de l'imprimante. L'inconvénient d'avoir à monter plusieurs disquettes peut être éliminé en achetant un disque plus grand, ou des unités de disques multiples ou à double densité. Ces choix sont décrits dans le chapitre sur les périphériques.

De nouveaux disques à tête fixe bon marché sont devenus disponibles et devraient éliminer l'inconvénient des disquettes multiples à l'avenir.

## Les logiciels disponibles

La plupart des systèmes microordinateurs disponibles aujourd'hui offrent les ressources matérielles nécessaires pour exécuter toutes les opérations qui ont été décrites à une vitesse suffisante, à condition qu'il n'y ait pas de calculs mathématiques complexes.

Malheureusement, *presque aucun système de gestion qui automatise complètement toutes les tâches n'est encore disponible.* En effet, on peut voir d'après la description qui a été présentée, que la tâche du système de gestion de fichiers et des divers programmes de traitement est complexe. De telles facilités n'existent que sur les gros ordinateurs.

Le coût de développement de tels programmes est beaucoup plus important que le coût de développement du matériel sur lequel ils résident. Pour cette raison, les constructeurs ne sont pas encore prêts à investir dans ce domaine aussi longtemps que cela n'est pas nécessaire pour vendre leur matériel. Les sociétés de consultants ou de services sont plus à même de s'intéresser à la vente de ces programmes de gestion. Cependant comme le logiciel n'est pas légalement protégé contre la duplication de manière efficace, il est difficile de vendre des logiciels pour des matériels de faible coût d'une manière profitable, du moins pour le moment. (L'avènement d'un marché de masse changera ceci).

La plupart des systèmes microordinateurs disponibles aujourd'hui offrent quelques programmes pour la gestion.

Ils offrent en général un système général de gestion de fichiers, qui permet à l'utilisateur de créer et de manipuler des fichiers symboliques. *Ceci est totalement insuffisant.*

Des programmes spécialisés doivent être disponibles pour la capture automatique des données, leur mise à jour et la modification automatique des autres fichiers.

La plupart des programmes disponibles résolvent seulement *un* de ces problèmes. Ainsi il y a d'habitude un programme de comptes créditeurs, un programme *séparé* de comptes à payer et un programme inventaire *séparé*. Ces « packages » séparés sont utiles pour maintenir des fichiers indépendants. Si le nombre de transactions est élevé pour l'un de ces fichiers, ils peuvent fournir des services appréciables. Cependant, ils ne font qu'une partie du travail. L'utilisateur du système doit mettre à jour manuellement tous les fichiers qui sont impliqués dans une même transaction.

Maintenant que les limitations ont été soulignées, *y a t-il une raison de ne pas considérer un système microordinateur pour une application de gestion ?* Non.

Les possibilités que nous avons décrites sont celles qui seraient idéales pour un système de gestion automatisée. Toutefois, même des solutions éloignées de cet idéal fourniront des résultats dont la valeur excède largement le coût du système. Un système utile peut être acheté pour 5 000 à 10 000 F.

Un tel système permettra, avec un minimum de logiciel, d'automatiser l'inventaire, les comptes créditeurs, les comptes débiteurs et peut être les fichiers d'adresses, les commandes en attente, la paye et le calcul des impôts. De tels bénéfices compensent déjà à eux seuls le coût initial du système. Pour cette raison, un système microordinateur est un outil de valeur pour une automatisation *partielle* d'un système de gestion. Il faut cependant avoir à l'esprit qu'un coût caché devra être pris en compte.

Si des fichiers volumineux sont créés, ou si des programmes spécifiques doivent être écrits, l'investissement logiciel deviendra dominant. Au bout d'un certain temps, l'investissement nécessaire à la structuration de ces fichiers et au développement de nouveaux programmes devient plus important que le coût initial du système. Après quelques années, des limitations se feront sentir et il sera désirable de passer à un système plus important et plus complet.

Bien sûr il est possible que le constructeur du système original en ait étendu les possibilités. Toutefois, ceci est peu probable à long terme. Du fait du coût constamment décroissant du matériel, il est à prévoir qu'en quelques années le matériel lui-même sera devenu obsolète et que le constructeur vendra un nouveau système plus performant mais qui ne sera pas complètement compatible avec le premier. L'utilisateur commercial devra alors effectuer une restructuration de ses fichiers et de ses programmes pour passer au nouveau système.

Cependant, le sentiment de l'auteur est que, même si le système original est abandonné après quelques années, il permettra une

transition de grande valeur vers le monde véritable de la gestion informatique.

Il est exact que l'investissement matériel initial sera perdu au bout de quelques années, et qu'une partie appréciable de l'investissement logiciel le sera aussi. Cependant, cela se traduira en une restructuration des procédures commerciales, une formation de personnel à des procédures automatiques et une prise de conscience des responsables des possibilités désirables dans un système futur. Un tel système apportera des bénéfices initiaux immédiats qui compenseront largement les coûts, et il fournira des bénéfices substantiels au niveau de la formation du personnel. Après un an ou deux, l'acheteur intelligent qui comprend la nature de ses besoins, sera en mesure de choisir le système idéal pour ses besoins. A ce moment, son second choix sera très probablement le choix optimal.

La comparaison est analogue à la recommandation faite à un nouveau conducteur de commencer par conduire une voiture peu coûteuse avant d'acheter la voiture de ses rêves. La première voiture peut être inadéquate, mais joue le rôle de banc d'essai qui permet à l'utilisateur d'apprendre et de développer ses compétences.

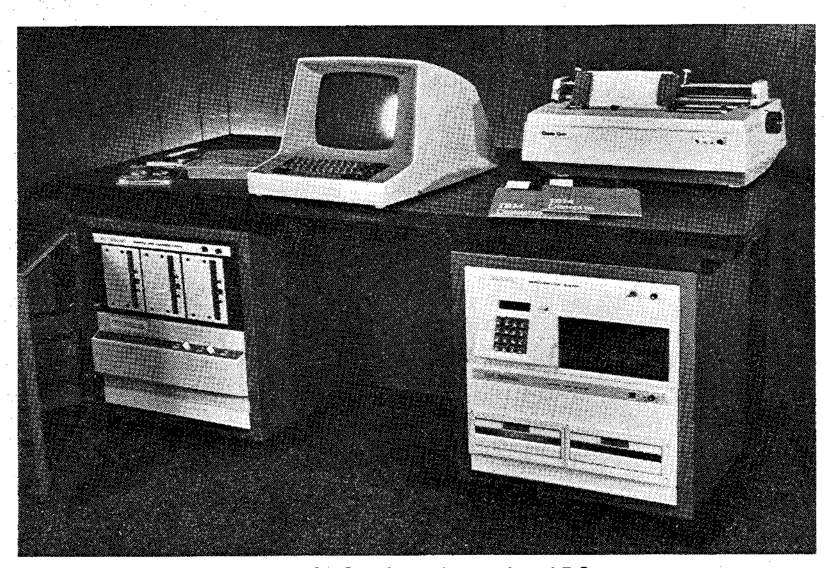


Fig. 7.8 : Système de gestion ADC.

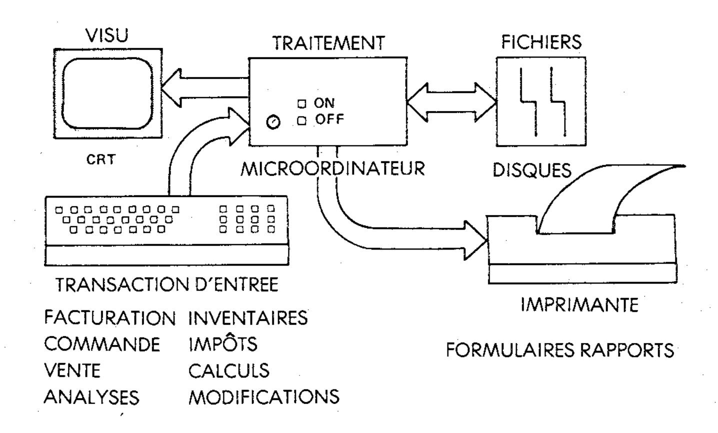



Fig. 7.9 : Les éléments d'un système de gestion.

Fig. 7.10 : Centre système de gestion standard ALTAIR

**Et un miniordinateur ?**

A première vue, les miniordinateurs, plus grands, plus puissants, plus coûteux, sembleraient apporter une solution au problème de la gestion commerciale par ordinateur. Ceci n'est pas nécessairement vrai. En effet, plus de 90 % des miniordinateurs sont utilisés pour des applications autres que la gestion. 10 % seulement des miniordinateurs sont utilisés pour la gestion. La plupart sont utilisés pour l'automatisation industrielle et l'acquisition de données spécialisées. La raison en est à nouveau la même : les constructeurs vendent du matériel. Ils ne sont pas intéressés par le développement d'un logiciel très coûteux sur lequel ils ne pensent pas réaliser de profits. Les meilleurs systèmes pour les applications de gestion, c'est-à-dire ceux qui incorporent un maximum de logiciel de gestion sont disponibles sur les gros ordinateurs qui coûtent plus de 100 000 F. Sur de tels systèmes coûteux, il devient raisonnable d'installer des logiciels aussi coûteux.

Toutefois, l'un des paramètres de l'équation a changé. La disponibilité de systèmes microordinateurs à très faible coût signifie que, pour la première fois, il devient possible de vendre du logiciel qui puisse être amorti sur un grand nombre d'unités.

Pour cette raison, on peut s'attendre à ce qu'une majorité des programmes hautement complexes disponibles jusqu'à présent seulement sur les gros ordinateurs, le seront désormais sur les microordinateurs.

On peut prédire que, en quelques années, les microordinateurs vont devenir des outils de traitement puissants pour la gestion, équipés de la plupart des fonctions nécessaires pour assurer une gestion commerciale efficace. Les systèmes disponibles aujourd'hui sont un premier pas dans cette direction.

**Les programmes de gestion sur mesure**

Une fois les besoins spécifiques d'une affaire identifiés, il devient possible d'évaluer si un système microordinateur spécifique répond à ces besoins.

Si ce n'est pas le cas, des programmes ou des packages supplémentaires devront être obtenus. La question essentielle est alors : *cela vaut-il la peine d'écrire un programme ou de traiter avec une société de service pour réaliser les programmes spécifiques nécessaires ?*. La réponse normale est : *non.*

Naturellement, si les programmeurs sont disponibles dans l'entreprise ou si la valeur éducative de la programmation vous semble importante, cela peut valoir la peine d'écrire vos propres programmes. Le plus important est que la plupart des utilisateurs de gestion *sous-estiment systématiquement la quantité de travail nécessaire pour obtenir un programme correct, utilisable et documenté.*

Il est en effet possible de développer rapidement dans un langage tel que BASIC un programme qui semble répondre aux besoins principaux. Malheureusement, à moins qu'un tel programme ne soit bien développé, bien documenté et adapté au système, il pourra faire plus de mal que de bien.

Ainsi, on pourra s'apercevoir qu'après avoir rentré plusieurs items dans un fichier, le programme devient essentiellement inutilisable, parce que son efficacité chute brusquement à partir d'un certain point ou tout simplement parce qu'il est incapable de dépasser cette limite.

On peut également s'apercevoir que lorsque certains types de données sont entrées ou utilisées, des défaillances incompréhensibles se produisent, auxquelles personne au sein de la société ne sait remédier.

La création d'un système complet, sans erreurs et véritablement utilisable est normalement une tâche longue et coûteuse.

A moins que les coûts de développement élevés d'un tel programme ne soient clairement justifiés par les bénéfices, *il vaut en général, mieux ne pas envisager de développement sur mesures.*

Les programmes qui existent depuis un certain temps et qui ont été testés sur d'autres utilisateurs sont les plus intéressants à acquérir. De plus, un système doit être *intégré* d'un point de vue logiciel. Tous les programmes devraient être capables de traiter des fichiers communs et de mettre à jour les fichiers requis. Des programmes additionnels qui seraient simplement ajoutés au système pourraient ne pas être capables d'utiliser ces facilités ou pourraient créer des structures que d'autres programmes ne pourront pas utiliser.

Cependant les programmes tout prêts (« packages ») pourraient ne pas être optimaux pour vos problèmes. Dans ce cas vous devrez évaluer un compromis :

- les programmes tout prêts peuvent être modifiés ou adaptés, de préférence par un programmeur au sein de l'entreprise. Ceci

résultera en une meilleure efficacité des programmes pour les procédures ou besoins spécifiques particuliers à votre entreprise.

- l'adaptation ou l'ajout de programmes accroîtra le coût total du système et peut résulter en une fiabilité diminuée.

## RECAPITULATION

La plupart des systèmes microordinateurs sont disponibles aujourd'hui avec des programmes simples de gestion. Ces programmes sont conçus pour *automatiser les transactions*. Chaque entreprise avec un grand nombre de transactions répétitives retirera un bénéfice de l'automation par ordinateur.

Cependant ceci n'est vrai que si le package standard disponible avec le système convient à l'entreprise. Par exemple, un programme d'entrée de ventes peut poser tellement de questions qu'il demandera un temps double de celui d'un traitement manuel. (Ceci arrive souvent). Dans ce cas, l'automatisation n'est avantageuse que si d'autres bénéfices en résultent.

De tels bénéfices supplémentaires sont : l'application forcée d'une discipline rigoureuse d'entrée des données, le traitement automatisé, la mise à jour automatique de plusieurs fichiers, l'établissement de rapports et journaux.

Si ces avantages (et ces programmes) supplémentaires existent, alors même un programme d'entrée des données non-optimal, encombrant et « standardisé » fournira un bénéfice.

Ceci sera toujours la supposition que nous ferons.

**Attention :** si ces facilités supplémentaires n'existent pas encore sur le système que vous envisagez d'acquérir, vous devriez évaluer soigneusement avantages et inconvénients. De nombreuses petites entreprises trouvent le traitement manuel des commandes, et la tenue manuelle de la comptabilité plus pratiques et plus maniables qu'un programme commercial tout-prêt « standard ».

Si un programmeur à temps partiel est disponible, ou si sa présence peut être justifiée, de nombreux programmes spécialisés peuvent être développés qui conviendront exactement à la société. Si le programmeur est bon, tous les bénéfices possibles liés à un ordinateur en résulteront, et les bénéfices indiqués ci-dessus en résulteront : efficacité, moindre coût, fiabilité accrue, gestion instantanée de l'information, paiement plus rapide des comptes dûs, automatisation de techniques de gestion nouvelles (commandes de

renouvellement automatiques, vérification du crédit, mailings sélectifs).

Les microordinateurs disponibles aujourd'hui disposent de toutes les ressources matérielles requises à un coût de moins de 50 à 100 000 F. Le logiciel est la clé du problème.

## ACHAT D'UN SYSTEME : UNE RECAPITULATION

### 1- Il doit y avoir un besoin.

La nature et la quantité de travail doivent justifier le changement. Ceci peut être mesuré par le nombre de transactions ou le nombre de rapports identiques à engendrer. De plus, un système ordinateur peut être justifié par les problèmes uniques qu'il peut résoudre dans une situation spéciale, telle que la gestion de l'inventaire, celle du fichier d'adresses, ou les rapports de gestion.

### 2- Quand acheter ?

Les prix de l'électronique continuent à baisser chaque année, tandis que le prix des périphériques est généralement stable ou légèrement décroissant.

Le système de demain sera toujours moins cher que celui d'aujourd'hui. Cependant, considérons ceci : l'achat d'un ordinateur peut résulter en une économie de N francs tous les mois. Différer la décision de m mois représente donc une perte de m x N francs.

### 3- Lequel acheter ?

Les options principales et les équipements disponibles vont maintenant être présentés.

### Autres précautions à prendre :

— Prévoir une copie de secours des fichiers tous les jours sur disque, bande ou papier, de façon que tout fichier perdu ou endommagé puisse être reconstitué.

— Se protéger contre les fluctuations du courant secteur, si le problème se pose dans votre région.

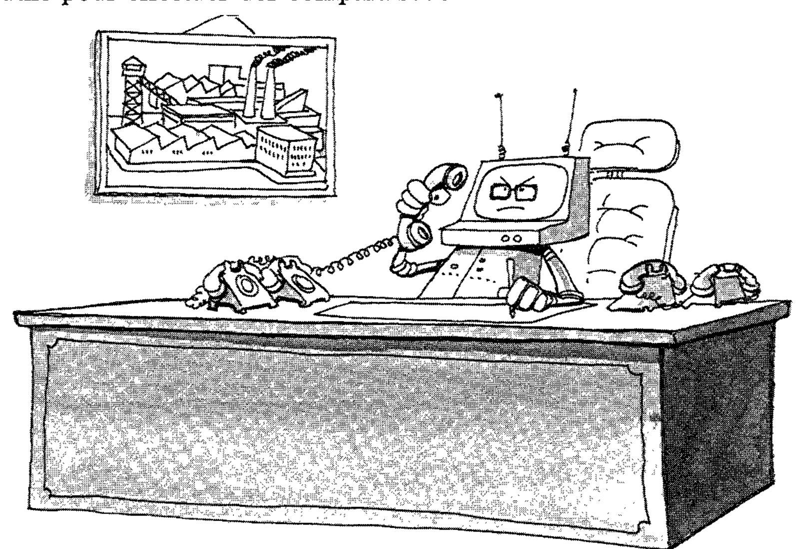
## RECAPITULATION

Les microordinateurs ont été classés en trois catégories :
— Simple carte (formation)
— Système personnel (jeux et traitements limités)
— Système général.

Dans la dernière catégorie il existe un grand nombre de microordinateurs disponibles aujourd'hui qui offrent des combinaisons diverses de ressources à l'intérieur de la boîte.

**Pour l'utilisateur de gestion :** seuls sont à considérer les microordinateurs disposant des caractéristiques suivantes :
— Mémoire complète (extensible jusqu'à 64 K)
— BASIC complet (de préférence résident en ROM pour l'efficacité d'exécution)
— Ensemble complet de périphériques avec des bus standards pour connecter d'autres périphériques
— Logiciel complet incluant un DOS (système opératoire pour disque) et système de fichiers
— Disponibilité de packages de gestion qui peuvent s'exécuter directement sur ce processeur.

**Pour l'utilisateur personnel :** la plupart des systèmes offrent des performances similaires, mais différents suivant leurs options d'entrée/sortie et leur extensibilité. L'analyse ci-dessus devrait être utile pour effectuer des comparaisons.



# LISTE DE CONTROLE
# DES FACILITES SOUHAITABLES EN GESTION

## 1. LISTES

- ☐ Clients
- ☐ Fournisseurs
- ☐ Inventaire
- ☐ Commandes
- ☐ Vendeurs
- ☐ Liste d'expédition
- ☐ Liste de ramassage
- ☐ Employés

## 2. JOURNAUX

- ☐ Paiements reçus
- ☐ Général
- ☐ Comptes débiteurs
- ☐ Comptes créditeurs
- ☐ Banque
- ☐ Impôts
- ☐ Taxes

## 3. FORMULAIRES COMMERCIAUX

- ☐ Factures
- ☐ Etiquettes
- ☐ Chèques
- ☐ Paie
- ☐ Lettres standard
- ☐ Rappels
- ☐ Notices spéciales
- ☐ Déclarations fiscales

## 4. RAPPORTS

- ☐ Comptes en retard
- ☐ Rapports de ventes
- ☐ Rapports d'inventaire (par âge, catégorie, item, etc.)
  Commandes de rappel
- ☐ Comptes classés par âge (débiteurs, créditeurs)
- ☐ Fichiers classés
- ☐ Ventes clients
- ☐ N'importe quel fichier classé par critère
- ☐ Commission/performance des vendeurs
- ☐ Vacances, absences, primes du personnel.

# CHAPITRE VIII
# CHOIX D'UN SYSTEME



# CHOIX D'UN SYSTEME

## INTRODUCTION

On peut être guidé, dans le choix d'un système microordinateur, par les mêmes principes généraux que pour le choix de tout système de haute complexité. Il existe de nombreux compromis possibles qui conduisent à une large gamme de performances, de commodités et d'avantages ou d'inconvénients. Pour faire un choix raisonnable, l'utilisateur doit d'abord définir (et comprendre) ses objectifs principaux. Une fois ceux-ci établis, on peut avancer des critères d'évaluation raisonnables. Comme la plupart des utilisateurs espèrent en fait que leur système sera le meilleur à tous les points de vue (et bien sûr bon marché), il peut y avoir de sérieuses déceptions. Dans la première partie de cette section, on va présenter les critères d'évaluation usuels, avec une analyse des critères possibles. Dans la deuxième partie, une synthèse présentera les types de systèmes les mieux adaptés à des classes d'applications spécifiques. Enfin, le chapitre 10 présentera une revue des principaux systèmes existant sur le marché.

## CRITERES DE CHOIX

Les principaux critères de choix sont d'habitude : un prix bas, la puissance, un matériel complet, un logiciel complet, la commodité et la fiabilité. Passons-les chacun en revue.

### Le prix

D'un point de vue général, la guerre des prix entre les constructeurs est telle que tous les systèmes équivalents se vendent pratiquement au même prix pour la même configuration. On ne peut donc pas appliquer le critère du prix à un constructeur donné, mais ce critère tend à être significatif pour un sous-ensemble de composants ou de fonctions d'un système donné à un moment donné. Naturellement, à tout moment, la guerre des prix peut

rendre un système donné plus attractif que les autres pour un temps limité.

En règle générale, l'utilisateur qui recherche le prix le plus bas possible peut d'abord envisager un système sur une seule carte. Pour moins de 1 500 F il peut acquérir un système minimal à microprocesseur qui n'exige en supplément qu'une alimentation coûtant environ 300 F. On communique avec un tel système en hexadécimal. Si l'utilisateur a d'autres ambitions et veut faire de la véritable programmation, il aura besoin d'un système possédant au minimum : une carte avec au moins 4 K de RAM, un coffret, une alimentation, un clavier alphanumérique, une unité d'affichage à écran pour les sorties et une unité de cassettes comme mémoire de masse.

Toutefois, il faut insister sur le fait que si l'utilisateur songe à un système plus complet, le coût de la carte unité centrale n'est plus le coût dominant dans le total. Cela a toujours été vrai dans l'histoire des ordinateurs. *Le coût principal dans un système est celui des périphériques*. Une bonne imprimante est à elle seule plus chère que l'unité centrale. En outre, la valeur totale d'un système ne peut se mesurer uniquement en termes de matériel et il faut insister sur le fait que le logiciel représente habituellement une composante très chère de tout système.

Pour cette raison, on peut dire que, sauf pour une dépense vraiment minimale indépendante de tout autre critère, le coût du coffret microordinateur ne constitue pas un critère de choix significatif.

**Evolution des prix**

Chaque année, les prix de presque tous les composants électroniques d'un système diminuent. Cela a toujours été vrai dans l'histoire des ordinateurs et la limite n'est pas encore atteinte. En revanche le prix des périphériques risque de ne pas décroître de façon significative, en raison de leur importante partie mécanique.

A tout moment, une fois qu'un choix a été fait et que le système a été livré, on peut affirmer qu'il existera très bientôt un choix moins cher pour au moins un des éléments du système !

Cela ne signifie pas qu'on n'a qu'à attendre. Faites le meilleur choix maintenant et commencez à utiliser le système. Le plaisir — si c'est un ordinateur personnel — ou les bénéfices — si c'est un

système de gestion — commenceront immédiatement et ils seront plus importants que toute économie obtenue en ne faisant rien.

En outre, le gain de connaissances que vous réaliserez en utilisant un ordinateur sera profitable dans votre vie professionnelle et pour le choix de systèmes futurs.

En bref, une fois que vous avez déterminé que le niveau de prix du système en vaut la peine, ne perdez pas de temps à le réduire de 10 ou 20 %.

**Performances**

Les soi-disant « performances » peuvent être extrêmement trompeuses. Voyons d'abord les performances du matériel. Elles se ramènent souvent à la vitesse élevée de l'unité centrale qui doit être couplée à un temps d'accès de la mémoire assez court pour que le microprocesseur n'ait jamais à attendre les données.

Tout d'abord, on a tendance dans les publicités à relier a priori la vitesse du microprocesseur et la fréquence d'horloge : « ce microordinateur a une fréquence d'horloge de 2 MHz », « tel autre a une fréquence de 4 MHz ». Cela veut-il dire que le second processeur est deux fois plus rapide que le premier ? Pas nécessairement. En effet il n'y a pas de relation simple entre la fréquence d'horloge et la vitesse d'exécution des instructions. La fréquence d'horloge donne le rythme suivant lequel le microprocesseur doit recevoir des impulsions pour exécuter les instructions. Mais l'exécution d'une instruction exige *plusieurs* périodes d'horloge. Il n'y a pas deux microprocesseurs qui utilisent le même nombre d'impulsions par instruction. Pour cette raison la vitesse d'horloge des différents microprocesseurs est une mesure fallacieuse de leur rapidité. Naturellement, la comparaison de deux vitesses d'horloge pour le même microprocesseur est significative. On peut comparer ainsi un Z80 à 2 MHz et un à 4 MHz. Toutefois il n'est pas évident que le Z80 à 4 MHz aille vraiment deux fois plus vite que le Z80 à 2 MHz. C'est parce qu'il faut rechercher les données et les instructions en mémoire. Il faut donc que la mémoire du système 4 MHz soit deux fois plus rapide que celle du 2 MHz ce qui n'est pas toujours le cas (la vitesse de la mémoire doit être en accord avec celle du processeur).

Voyons maintenant les performances de la mémoire. Elles sont habituellement caractérisées par le temps d'accès de la RAM, c'est-à-dire le temps qu'il faut pour chercher un mot dans la RAM. Plus la

RAM est rapide, plus elle est chère. Comme leur technologie évolue constamment, il est difficile de préciser la vitesse typique d'une RAM aujourd'hui puisqu'il est pratiquement certain que, dans un an, il y en aura une plus rapide, au même prix ou même moins chère. Dès que la RAM est assez rapide pour que le microprocesseur reçoive des données ou des instructions sans délai, il n'y a rien à gagner à utiliser une mémoire plus rapide. Quand on évalue les performances du système, il faut supposer que la vitesse de la RAM est suffisante pour le microprocesseur. Il est intéressant de constater que quelques systèmes de bas prix utilisent une mémoire lente qui ralentit un microprocesseur rapide. Ici nous supposerons que la plupart des systèmes considérés auront une mémoire dont la vitesse est adaptée au microprocesseur considéré. Dans ce cas les performances de la mémoire n'influent pas sur le choix. Nous venons de perdre un critère de sélection. Comment allons-nous mesurer les performances du matériel ?

**Programmes de test de performances**

La seule véritable manière de mesurer les performances hardware d'une unité centrale est de faire tourner des programmes de test de performances, ou bancs d'essai. Les programmes de banc d'essai sont des programmes qu'on peut qualifier de « typiques pour l'application considérée ». Un programme de test doit être passé sur différents ordinateurs, après quoi on peut faire des comparaisons de temps. Il n'existe pas d'instructions typiques pour tous les ordinateurs ; par suite, la comparaison d'instructions spécifiques entre elles peut être complètement fausse. L'ennui est qu'il n'existe pas de banc d'essai standard non plus. Pour des applications spécifiques telles que le transfert de données en parallèle, on peut écrire un programme correspondant et donc établir un banc d'essai. Dans la plupart des cas toutefois, la gamme d'applications à envisager est si vaste qu'il est très difficile d'écrire un programme de test de performances raisonnable. En outre, un banc d'essai écrit par un constructeur est presqu'inutile. *Pour être valable, un programme de test doit être écrit par l'utilisateur du système lui-même,* puisque c'est lui qui programmera le système. Le constructeur peut introduire des astuces en y passant des mois ou des années, de sorte que son programme de banc d'essai tourne beaucoup plus vite sur son processeur que sur celui des concurrents. Mais un tel programme n'est pas « typique » pour l'utilisateur.

Néanmoins, hormis quelques programmes d'applications spécifiques, il n'y a pas de « programmes moyens ». Certains programmes font beaucoup de calcul, d'autres font surtout des entrées-sorties. Il n'y a par suite pas de banc d'essai standard.

Si l'on cherche à évaluer un système d'usage général, les programmes test ne peuvent être utilisés que comme un critère d'évaluation grossier des performances du système. Nous avons presque perdu notre troisième critère, qui était semble-t-il le meilleur. Quel critère utiliser alors ?

**Les critères importants**

Disons tout d'abord que la vitesse pure du microprocesseur n'est pas le critère le plus significatif pour les performances du système global.

Ensuite, la plupart des microprocesseurs aujourd'hui sur le marché ont pratiquement les mêmes performances. On peut arguer que les plus rapides vont souvent deux fois plus vite que leurs concurrents. Mais cela n'est significatif *que si le logiciel du système est efficace. Le seul critère fondamental pour le fonctionnement d'un système destiné à être programmé en langage évolué est l'efficacité du logiciel.* Même si l'on doit programmer en assembleur, l'habileté du programmeur peut avoir sur la vitesse globale du système une influence plus marquée que les performances du matériel. En outre, il est probable que d'ici un ou deux ans, des boîtiers plus rapides seront disponibles de sorte que l'on pourra améliorer la vitesse à moindre coût en remplaçant une ou plusieurs cartes. Comme d'habitude, la vitesse du logiciel est le point-clé du rendement du système.

Enfin, le goulot d'étranglement pour les performances d'un système microordinateur à usage individuel ou professionnel, ne se situe pas en général au niveau du traitement, mais à celui des entrées-sorties. On a besoin d'un disque rapide et d'une imprimante rapide : or chacun d'eux est beaucoup plus cher que l'unité centrale complète. Si le coût doit demeurer bas, ce sont donc eux qui limitent généralement les performances du système.

Les périphériques représentent généralement le facteur qui limite les performances pour les traitements orientés fichiers (applications de gestion).

La plupart des applications personnelles et de gestion sont écrites en langage de haut niveau tel que BASIC. Dans le cas d'un langage

évolué, l'efficacité de l'interprêteur (ou du compilateur) est cruciale. Par exemple, certains interprêteurs BASIC parmi les plus mauvais sont *50 fois plus lents* que les meilleurs. Lorsque l'on compare les microprocesseurs eux-mêmes, on trouve habituellement des rapports de 1 à 2 dans les performances alors que dans des comparaisons de logiciel, des rapports de 1 à 50 ne sont pas rares.

Qui a le meilleur logiciel ? Malheureusement, la réponse à cette question varie très vite. Il est vrai qu'au moment où nous écrivons, certains constructeurs ont des interprêteurs de toute évidence meilleurs que d'autres. Mais les interprêteurs sont sans cesse améliorés de sorte que la situation peut changer rapidement. Tout classement serait trompeur. Régulièrement, les revues pour amateurs (listées dans ce livre) publient des comparaisons entre divers interprêteurs ou autres programmes, établies à la suite des bancs d'essai. Le lecteur intéressé est invité à s'y reporter.

*En bref, le lecteur qui recherche des performances pour son système, est fermement invité à vérifier la rapidité des outils logiciels qu'il compte utiliser.*

**Matériel complet**

Un système de base est composé de :

— Un coffret microordinateur, avec possibilité d'extensions mémoire, des emplacements pour des interfaces périphériques supplémentaires, et une alimentation suffisante pour permettre à l'ensemble de fonctionner de façon fiable.

— Un clavier d'entrée avec une disposition de touches convenable.

— Une sortie sur écran.

— Une sortie permanente (imprimante).

— Une mémoire de masse — au minimum sur cassettes, de préférence sur disques.

Si le système fait appel au bus S100 (devenu un standard de cette nouvelle industrie), le fait que le matériel proposé soit complet n'a pas d'importance puisque les éléments sont interchangeables d'un constructeur à l'autre.

Si ce n'est pas le cas, la disponibilité de tous les éléments est cruciale surtout si l'on envisage des développements ultérieurs. Le bus S100 et les autres bus seront décrits plus loin.

L'acheteur prudent devrait toujours supposer qu'il n'y aura jamais d'autres options ou périphériques que ceux qui existent

réellement aujourd'hui. « Réellement » veut dire qu'un système en démonstration à une exposition peut être « factice ».

Enfin il peut y avoir des besoins spéciaux tels que nécessité d'une carte arithmétique flottante, ou besoin d'une interface à un terminal spécifique.

Les caractéristiques de tous les périphériques habituellement utilisés dans un système seront présentés et évalués en détail dans le chapitre suivant.

**Logiciel complet**

Un système, aussi beau qu'il soit, est inutile s'il n'est pas muni de toutes les facilités en logiciel requises par votre application.

Il doit comporter un moniteur, un éditeur et un processeur de langage tel qu'un interprêteur BASIC (voir le chapitre sur la programmation pour plus de détails).

Le logiciel revient très cher à développer et beaucoup de constructeurs s'efforcent de fournir le minimum qui leur permette de s'en tirer.

**Pour des applications de gestion,** on peut appliquer les critères suivants :

1. Disponibilité de programmes de gestion prêts à l'emploi, tournant sur votre système quel que soit le langage.
2. Un BASIC complet (voir le chapitre sur BASIC pour plus de détails).
3. Un bon système de fichiers pour la gestion automatique de vos fichiers.

**Pour un système d'enseignement :**

1. Disponibilité d'un BASIC complet.
2. Disponibilité d'un logiciel de temps partagé permettant à plusieurs utilisateurs d'accéder au système simultanément.

**Pour un système personnel :**

1. Un BASIC complet.
2. Un bon système de fichiers.
3. Un assembleur.

Ce dernier point peut paraître évident. Mais, de fait, certains « ordinateurs personnels » ont un BASIC acceptable mais n'ont pas d'assembleur acceptable, ou ne permettent pas de faire le lien entre les deux, ce qui empêche d'utiliser commodément des sous-

programmes assembleurs. La présence de l'assembleur n'est pas indispensable toutefois si l'on ne prévoit que de programmer en BASIC.

En outre, et dans tous les cas, il devrait y avoir un éditeur puissant et complet permettant de modifier facilement les programmes (voir le chapitre sur la programmation pour plus de détails).

**Commodité**

La commodité est une des clés de l'utilisation d'un système avec succès. La plupart des utilisateurs ne se soucient pas de réinventer la roue, ni même d'avoir à créer des interfaces complexes ou de reprogrammer des algorithmes qui ont été découverts depuis longtemps. Pour cette raison, les microprocesseurs qui étaient les plus utilisés dans les applications antérieures sont aussi les plus utilisés dans les ordinateurs personnels. La plupart des microordinateurs font appel à l'Intel 8080, au Zilog Z80, au Motorola 6800 et au MOS Technology 6502. Il est intéressant de noter que Zilog a été fondée par des transfuges d'Intel, de même que MOS Technology l'a été par des transfuges de Motorola. A cause de la profusion de software disponible pour le 8080 ou le Z80 (ils ont un jeu d'instruction compatible) aussi bien que pour le Motorola 6800 et — dans une moindre mesure — pour le MOS Technology 6502, ces microprocesseurs sont maintenant les plus largement utilisés dans les microordinateurs personnels.

La commodité implique que toutes les fonctions nécessaires pour mettre le système en œuvre immédiatement soient disponibles et que les efforts et le temps nécessaires soient minimaux.

De façon spécifique, la commodité signifie :

1. Système complet prêt à l'emploi du point de vue matériel et logiciel.

2. Facilité de mise en œuvre. Ceci veut dire principalement aides logicielles puissantes (éditeur, interprêteur, système de fichiers), et matériel en rapport (disque rapide, imprimante rapide).

3. Disponibilité et applicabilité immédiates.

4. Excellente documentation. C'est là souvent un facteur-clé pour l'utilisation efficace de votre temps.

**Performance globale du système**

S'il est probable que le système doive manipuler des fichiers

plutôt que d'exécuter des programmes qui traitent des données résidant en mémoire centrale, alors l'efficacité de l'unité de mémoire de masse est cruciale pour les performances du système. En particulier, pour des applications de gestion, les fichiers seront volumineux et ne peuvent tenir simultanément dans la mémoire centrale du système qui est limitée à 40 ou 48 K (généralement 16 K sont occupés par les programmes en ROM). La vitesse du disque — souple ou rigide — devient cruciale. En fait, le temps d'accès du disque est plus important que la vitesse d'exécution du processeur dans les cas où l'on a besoin d'accès répétés à des fichiers (particulièrement dans les applications de gestion).

Si l'on a besoin d'imprimer de nombreux fichiers, une imprimante rapide devient nécessaire, sinon le fonctionnement du système se ralentit considérablement. Une imprimante rapide est généralement le périphérique le plus coûteux. Pour cette raison, on utilise souvent à sa place une imprimante meilleur marché, ce qui entraîne une inefficacité globale du système étant donné que l'utilisateur doit attendre qu'un fichier ait fini de s'imprimer pour pouvoir poursuivre son travail.

**Fiabilité**

La plupart des microordinateurs, au début de leur conception, n'étaient pas construits en fonction des mêmes normes de fiabilité que les ordinateurs classiques. Ils étaient construits dans l'enthousiasme par un ingénieur seul, ils trouvaient un marché par accident et ne pouvaient pas être améliorés ultérieurement puisque leur configuration avait été standardisée par le nombre même des ventes. Toutefois, la haute densité caractéristique des circuits LSI est telle que le nombre de composants dans un système microordinateur est beaucoup plus réduit que dans les miniordinateurs traditionnels, ce qui permet à des méthodes de conception qui auraient été qualifiées d'incertaines dans le passé de fournir des systèmes dont la fiabilité globale est comparable à celle des miniordinateurs traditionnels. Pour toutes les applications pratiques, la plupart des microordinateurs sont très fiables en environnement « normal » à partir du moment où ils ont fonctionné correctement pendant les 100 ou 200 premières heures. Ceci est appelé la « période de mortalité infantile » pendant laquelle se produisent la plupart des pannes qui devaient se produire. La probabilité de panne ultérieure est beaucoup plus faible. Un environnement « normal » signifie l'absence de variations

extrêmes de la température et de l'humidité. En outre, on rencontre quelquefois des problèmes de bruit si l'organisation du bus est mauvaise et de distribution des alimentations lorsqu'on ajoute des cartes supplémentaires.

Le principal danger vient du succès-même des systèmes microordinateurs. Pour pouvoir fournir des systèmes à prix de plus en plus bas, certains constructeurs ont à l'occasion recours à des fournisseurs de composants moins fiables. Il est possible d'acheter à bien plus bas prix, sur le marché « parallèle », des composants non testés. Cela pourrait très bien marcher à condition de tester les composants à l'arrivée. Par manque de temps cela peut ne pas être fait, et des composants potentiellement défectueux peuvent se trouver expédiés au sein d'un système. L'ennui est que les pannes peuvent se produire dans des circonstances peu reproductibles comme des températures extrêmes ou simplement par détérioration progressive du composant, et cela peut arriver après expiration de la garantie. Généralement, le constructeur remplacera quand même le composant défectueux. Cependant trouver la panne dûe au composant défectueux peut être une tâche longue et délicate. La fiabilité et la réputation du constructeur constituent par conséquent un critère de sélection significatif pour se protéger contre ce genre de danger.

## TYPES DE SYSTEMES

### Kit, carte ou système ?

Les microordinateurs sont fournis sous deux formes principales : microordinateur sur une carte ou système complet présenté dans un beau coffret. En plus il y a généralement le choix entre le système tout monté ou le kit. Lequel choisir ?

Les avantages et les inconvénients de chacun sont simples : la carte a l'avantage du coût minimal et l'inconvénient de ressources hardware limitées. Le coffret est plus cher mais il offre les possibilités hardware d'un miniordinateur traditionnel. Examinons maintenant cette alternative plus en détail.

### Microordinateur sur une carte

Une carte-type comprend le microprocesseur plus les boîtiers de contrôle nécessaires, une quantité de mémoire limitée, telle que par

exemple 512 mots de RAM et 2 à 4 K de ROM, plus un minimum d'interfaces telles qu'une interface télétype, une interface cassette et des ports d'entrées-sorties d'usage général.

Vu sa structure standard, une carte peut facilement être fabriquée en grande série à bas prix. Toutefois, à cause de la surface limitée de la carte, il faut trouver un compromis entre les différents dispositifs qui pourraient être implantés sur celle-ci. Ces choix sont la quantité de mémoire, d'entrées-sorties, d'interfaces et la présence d'un clavier. On verra ci-dessous que la limitation de la mémoire est la plus gênante vu qu'elle exclut la possibilité d'utiliser un assembleur ou un langage évolué.



Trouver la bonne carte peut être difficile.

En raison de ces limitations, une carte peut servir de façon autonome à essentiellement deux applications.

*1. Comme unité de commande.* Une telle carte fournit la capacité de calcul d'un microordinateur complet pour 1 000 à quelques milliers de francs. Elle peut être utilisée pour commander directement des relais, des moteurs ou autres. C'est un automate de commande de bas prix, programmable, général, qui peut être utilisé

de façons multiples. La principale application est l'automatisation industrielle, et elle peut aussi servir à la maison ou au bureau pour contrôler des applications simples comme la régulation de moteurs, le contrôle de processus, la surveillance de l'environnement, des alarmes anti-vol, la minuterie automatique de l'éclairage ou une installation d'arrosage temporisée.

2. *Comme outil d'enseignement.* Une telle carte a grand intérêt, pour la compréhension de l'interconnexion d'un système microordinateur typique ainsi que pour apprendre à programmer en langage machine.

Toutefois, il n'est pas raisonnablement possible de développer des programmes longs et complexes à l'aide d'un microordinateur sur une carte. A cause de la taille-mémoire limitée, la carte ne peut généralement pas permettre d'utiliser un assembleur ou un interprêteur. Par suite, sur la plupart des cartes, on entre les instructions au clavier sous forme de suites de nombres en hexadécimal. C'est un procédé long et propice aux erreurs. La plupart des utilisateurs arrive à entrer de cette façon quelques dizaines d'instructions et ainsi à apprendre les bases de la programmation au niveau machine. Si l'on dispose d'assez de mémoire (généralement sur une carte extérieure), il est possible de faire tourner un assembleur. Cela autorise l'usage de programmes *symboliques*. Mais pour écrire des programmes symboliques, il faut un *clavier alphanumérique* ou une unité d'entrées-sorties standard telle qu'un *Teletype*. Si l'on doit dépenser 5 à 10 000 francs pour un télétype, il y a d'autres choix raisonnables qu'une carte unique pour l'unité centrale.

En résumé, la carte est un ordinateur bon marché qui peut être utilisée de façon efficace comme contrôleur d'application ou comme outil de formation pour comprendre la structure d'un microordinateur et apprendre les bases de la programmation. A titre d'exemple, une telle carte peut être utilisée à l'intérieur d'une machine de gestion mais pas seule.

Une carte est l'outil de formation le moins cher qui soit pour la formation personnelle.

## Le Système Microordinateur

Le système consiste en une ou plusieurs cartes réunies dans un coffret esthétique, plus la configuration minimale de périphériques

indispensables pour obtenir une utilisation raisonnable du système. Un système complet est aujourd'hui disponible pour 3 à 5 000 francs, c'est-à-dire le double du prix d'une carte unique. C'est pourquoi ils offrent une alternative d'apparence séduisante pour l'usage individuel. Un *système-type* comprend le coffret unité centrale plus un *clavier alphanumérique*, un *écran* et une *mémoire de masse*. Les systèmes les moins chers ont une mémoire de masse à *cassettes* tandis que les plus puissants ont des *disques souples*. En outre, pour les applications pratiques il faut une *imprimante*.

*En théorie un microordinateur peut faire tout ce qu'un ordinateur fait,* en se tenant dans ses limites hardware et software. Son inconvénient essentiel est d'être moins puissant qu'un miniordinateur traditionnel, c'est-à-dire plus lent et plus restreint en mémoire et possibilités d'entrées-sorties.

On peut distinguer deux types de systèmes : les systèmes de coût minimal et les microordinateurs d'usage général.

### Le système de coût minimal

Le système de coût minimal comprend le coffret unité centrale avec 4 K à 16 K mots de mémoire, l'alimentation, le clavier, un moniteur TV et une unité de cassettes (en option). Le coffret (souvent scellé) ne permet généralement pas d'ajouter de cartes supplémentaires au système.

Encore une fois, cette configuration a la même puissance de traitement nominale qu'un autre microordinateur mais elle est limitée par la taille mémoire et les périphériques. La mémoire peut généralement être un peu étendue et il y a souvent des périphériques supplémentaires disponibles. Cependant on ne dispose généralement pas du software capable de suivre. Le *système d'exploitation* requis pour gérer les différents périphériques et les différentes tailles mémoires doit être conçu correctement dès le départ pour les dimensions maximales ultérieures. Un système minimal est conçu comme tel (minimal). Il comprend un système d'exploitation minimal appelé moniteur qui gère le clavier et les cassettes. Le passage à une configuration plus étendue exige un système d'exploitation différent.

Le système minimal offre un assembleur et un mini-BASIC ou un autre interprêteur. L'utilisateur peut alors écrire des programmes symboliques, au niveau assembleur ou en BASIC (simplifié). En

outre il peut ranger et récupérer ses programmes sur cassettes, ce qui est commode.

Un tel système est normalement livré avec des cassettes contenant un choix de programmes tels que des jeux ou même des programmes « utiles » : gestion de compte-chèques, enseignement de l'arithmétique, etc.

Ce système est limité par sa mémoire et ses périphériques. Il peut servir à un usage personnel mais pas comme système de gestion d'une entreprise.

Ceci illustre encore une fois un fait qui a toujours été caractéristique de tous les ordinateurs : le coût de l'unité centrale n'est qu'une petite partie du coût total d'un système opérationnel. Le coût hardware dominant est, et a toujours été, celui des périphériques.

A titre de remarque finale, la carte unité centrale n'a pas besoin d'un grand coffret et elle est souvent logée dans le coffret du clavier ou celui de l'écran.

En résumé, les systèmes minimaux sont des jouets éducatifs agréables avec lesquels il est possible de jouer, de faire fonctionner de nombreux programmes éducatifs ou d'apprendre à programmer. Néanmoins, comme ils ont peu de mémoire et une unité de mémoire de masse lente, la cassette, ils ne peuvent être utilisés en gestion puisqu'ils ne peuvent fournir une gestion de fichiers efficace. Ils peuvent en fait être utilisés à un grand nombre d'applications personnelles. Si l'on achète de la mémoire supplémentaire, ils peuvent abriter un assembleur, un éditeur ou un interprêteur et alors permettre de nombreuses applications. Toutefois, il faut noter que certains de ces systèmes offrent la possibilité d'être transformés en microordinateur général. Enfin, même ces ressources limitées peuvent être suffisantes pour une seule tâche de gestion, simple et spécialisée.

**Le microordinateur d'usage général**

Le microordinateur général est fonctionnellement équivalent aux miniordinateurs qu'on connaissait. C'est un coffret contenant une carte processeur plus les cartes supplémentaires nécessaires à la mémoire et aux interfaces d'entrée-sortie. Le coffret n'est pas totalement plein, ce qui permet d'insérer des cartes supplémentaires pour étendre la mémoire jusqu'à 40 ou 48 K, et pour connecter un certain nombre de périphériques supplémentaires. Un *système* type

est livré avec pour périphériques une — ou mieux, deux — unités de disques souples, un clavier alphanumérique de bonne qualité, une unité d'affichage cathodique de bonne qualité. En outre, il faut adjoindre au système une imprimante convenable. Un tel microordinateur général doit aussi comprendre une alimentation *capable d'alimenter le nombre maximum de cartes qu'on peut installer*. Un tel système n'est plus bridé par les limitations qu'on a vues précédemment et il est capable d'exécuter tout programme qui ne surcharge pas ses capacités de calcul.

Pratiquement, le microordinateur peut se présenter sous différentes formes de coffrets et même être inclus dans le coffret du clavier ou dans celui de l'écran. Pour avoir plus de souplesse si l'on ajoute des cartes supplémentaires, on a le plus souvent un coffret séparé, de la taille d'un tuner haute-fidélité.

Les avantages par rapport au miniordinateur traditionnel sont :

— Prix plus bas (peut-être dix fois moins)

— Moindre encombrement

— Moindre consommation et moindre dissipation : il en résulte que l'alimentation est plus petite et moins chère et on a besoin de moins de ventilateurs pour le refroidissement. Cela permet aussi une présentation élégante dans un style proche des appareils haute-fidélité qu'on a à la maison.

— Disponibilité de périphériques récents bon marché.

Les principales limitations si on le compare à un miniordinateur sont :

— Puissance de traitement plus faible.

L'utilisation d'un microprocesseur 8 bits entraîne une vitesse d'exécution plus lente. Cependant, sauf dans le cas de calculs arithmétiques complexes, cette limitation n'est généralement pas ressentie. Le bas prix du microordinateur autorise de l'employer à une tâche unique, éliminant la nécessité d'une gestion de tâches complexe ou d'un logiciel de temps partagé.

La puissance de traitement d'un microordinateur est suffisante pour la plupart des applications personnelles et de gestion.

— Moins de software.

Comme les microordinateurs sont plus récents, ils ont une bibliothèque de programmes moins riche que les miniordinateurs traditionnels. Mais l'écart est en train de se réduire rapidement et bientôt une grande quantité de logiciel sera disponible.

— Moins de possibilités d'entrées-sorties.

Le microordinateur a une gamme d'entrées-sorties limitée par rapport à un mini. Mais, là encore, ceci a une faible incidence pratique dans la plupart des utilisations.

En résumé, les microordinateurs offrent la plupart des caractéristiques des ordinateurs classiques si ce n'est qu'ils calculent plus lentement, mais coûtent une fraction de leur prix. Ils n'élimineront pas les minis puissants mais ils ont simplement ouvert un nouveau marché où l'on peut dédier un microordinateur à une tâche particulière. Cependant, dans la plupart des utilisations pratiques, ils apportent à tout utilisateur essentiellement la capacité de traitement d'un mini pour une fraction du coût.

## Faut-il un panneau de commande ?

Certains coffrets de microordinateurs sont livrés avec un panneau de commande (« panneau avant »), d'autres sans. Le panneau de commande est simplement une rangée de lampes et une rangée d'interrupteurs plus quelques lampes, interrupteurs et boutons. Il est conçu pour faciliter le dépannage à la main lorsqu'on programme en langage machine. Il affiche sous forme visible le contenu binaire des bus et des registres. Pour des applications de contrôle de processus qui sont généralement programmées au niveau assembleur, la présence d'un panneau de commande est bien vue de la plupart des programmeurs qui sont habitués au matériel, car il permet de visualiser instantanément l'état des registres ou des bus, sans avoir besoin d'une série d'instructions pour obtenir les informations. Cependant, le panneau de commande coûte cher, car il nécessite une carte de commande supplémentaire plus un programme spécial inclus dans le système. Il peut donc ajouter sensiblement au coût d'un petit système. C'est pourquoi, aujourd'hui, la plupart des microordinateurs sont livrés sans panneau de commande. Si l'on a l'intention d'utiliser le système le plus souvent au moyen d'un langage évolué, le panneau de commande ne sert simplement à rien. Même si l'utilisateur a l'intention de programmer en assembleur, l'absence de panneau de commande peut ne pas apporter d'inconvénient important. Quant à l'utilisateur qui n'a jamais vu de panneau de commande, il ne saura jamais de quoi il est privé. Et ce dont il est privé n'est pas très important puisque toutes les possibilités de diagnostic peuvent être également obtenues à partir de la console.

Les premiers microordinateurs (Altair, Imsai) étaient munis d'un panneau de commande puisqu'on s'attendait à ce qu'ils soient utilisés dans l'environnement de laboratoire traditionnel (et parce qu'ils avaient été conçus dans un tel environnement). Désormais, il n'y a généralement plus de panneau de commande. Il est inutile sur une machine de gestion. Si l'utilisateur programme en BASIC, il ne se servira jamais du panneau de commande.

Il faut insister sur le fait que l'absence de panneau de commande n'empêche pas de vérifier le contenu des registres internes. Cela devient simplement moins direct et la visualisation se fait à partir de la console (écran ou télétype).

Enfin, un système sans panneau de commande n'a plus qu'un bouton marche-arrêt, il est plus « élégant » et plus simple à utiliser et est, par suite, plus attrayant.

## Quel bus ?

La plupart des microordinateurs réalisent aujourd'hui leurs connexions internes au moyen d'un « bus » standardisé. Ce bus a entre 50 et 100 lignes et il transmet les signaux nécessaires entre les cartes du système. Le plus répandu est le bus S100 introduit à l'origine par Altair, puis utilisé par Imsai et beaucoup d'autres depuis. Il est basé sur le 8080 et peut être utilisé avec un Z80. Il est difficile à utiliser de façon raisonnable avec un 6800. (Il existe toutefois des adaptations du 6800 et du 6502 au bus S100). Il existe divers autres bus tels que le S50 (pour 6800), et d'autres. Leurs mérites relatifs sont comparés dans la réf. C 5. Généralement, si l'on recherche la simplicité et la commodité pour relier des dispositifs additionnels, le bus S100 a l'avantage d'être de loin le plus répandu. La plupart des cartes habituelles sont maintenant disponibles avec un interface au bus S100 et quelquefois à d'autres bus. Toutefois, même si le choix est plus restreint dans le cas d'un bus autre que le S100, cela n'apporte souvent pas de contrainte notable vu l'étendue du choix des cartes disponibles aujourd'hui.

Pour simplifier, si l'on considère que la souplesse maximum dans le choix d'un périphérique est un avantage important, le bus S100 a un intérêt notable. Cependant, un « bus standard » peut ne pas influer de façon majeure sur le choix d'un système dès lors que le système complet est disponible chez le constructeur, (s'il l'est).

## Les connecteurs au dos du coffret

Trois connecteurs au moins sont désirables :

*1. La boucle de courant 20 mA pour télétype*

C'est un port d'entrée-sortie série classique pour TTY ou autre terminal lent. Divers connecteurs à 4 broches sont utilisés.

*2. L'interface standard RS232C*

Indispensable. Ce connecteur peut servir avec n'importe quel périphérique muni d'un interface EIA (RS232C). Il servira presque toujours pour l'écran. On peut relier en parallèle plusieurs terminaux sur le même port.

*3. Interface parallèle pour imprimante*

Sortie parallèle 8-bits spécialisée pour imprimante rapide.

*Attention :* la plupart des microordinateurs ont un seul port d'entrées-sorties *série*. Dans ce cas le connecteur RS232C et le connecteur boucle 20 mA lui sont tous les deux rattachés, en parallèle.

Cela signifie que deux périphériques tels qu'un écran et un télétype peuvent être *reliés*, chacun sur un connecteur, simultanément, mais ils ne peuvent pas généralement être *actifs* simultanément. Il est clair qu'en entrée, les deux claviers ne doivent pas être utilisés en même temps.

En sortie, ils peuvent rester connectés simultanément s'ils fonctionnent tous les deux à la même vitesse (110 baud pour un TTY). Mais un terminal à écran est généralement réglé à une vitesse bien plus élevée (jusqu'à 9 600 baud).

L'existence des deux connecteurs RS232C et 20 mA n'est donc qu'une commodité. Elle ne signifie pas qu'on puisse connecter et faire fonctionner deux terminaux à la fois (il faudrait pour cela *deux* ports série).

Toutefois, dès que le microprocesseur est muni d'un port parallèle, on peut faire fonctionner simultanément une imprimante rapide parallèle et un organe sériel.

## Système en kit ou assemblé ?

Un kit permet des économies non négligeables par rapport à un système tout assemblé. Si on a l'intention de faire des économies, on peut acheter un kit et la valeur éducative que représente l'assemblage du système, ainsi que la satisfaction psychologique, font qu'il vaille peut-être la peine d'essayer. Toutefois, pour un

microordinateur en coffret, il faut insister sur le fait que la qualité du montage est importante pour un fonctionnement fiable du système. C'est pourquoi il est probable qu'un utilisateur novice fera mieux d'acheter un système tout monté et de consacrer son temps à apprendre à programmer utilement son système plutôt que de passer des nuits à essayer d'isoler des composants ou soudures défectueux. Les économies qu'on peut réaliser avec un kit risquent de ne pas se réaliser si l'utilisateur ne dispose pas des équipements de test adéquats. Les appareils de mesure convenables peuvent être très chers et leur achat peut effacer complètement les avantages financiers qu'on attend du travail de montage. Ne pas disposer de tels outils de mesure peut mener à des erreurs de fonctionnement du système qui n'apparaîtront que plus tard, lors de conditions exceptionnelles ; cela ruine la valeur et la fiabilité du système.

Un autre problème est apparu récemment : celui des composants non fiables. Des composants rejetés ont été volés chez les gros fabricants des semi-conducteurs ; ils ont été re-étiquetés et sont apparus sur le marché grand-public.

On ne peut pas faire grand-chose contre cela. Le problème avec un kit est que s'il ne marche pas, l'utilisateur doit isoler le composant défectueux. Avec un système tout monté, la situation est meilleure. Mais des composants défectueux peuvent ne se manifester qu'un an après, ou seulement à température élevée.

Cependant, on doit porter au crédit des microordinateurs que :

1. Ils sont intrinsèquement plus fiables que les miniordinateurs traditionnels parce qu'ils ont beaucoup moins de composants.
2. Ils sont faciles à réparer et le remplacement de toute une carte ne coûte pas cher.

En résumé, soyez avertis mais ne vous effrayez pas.

Nous vous suggérons d'apprendre à programmer plutôt qu'à faire des soudures. Mais les économies dues à un kit peuvent vous rendre accessible un système que vous ne pourriez pas vous payer tout monté.

## RECAPITULATION

On a vu un certain nombre de critères de choix, dont l'importance a été évaluée tour-à-tour. Que reste-t-il ? Principalement le *support logiciel.*

Le support logiciel est probablement le point-clé dans le choix d'un système. Si un système doit être utilisé pour une variété

d'applications, la plupart des utilisateurs ne se soucient pas de reprogrammer le système à chaque fois. Ils veulent des programmes qu'ils peuvent charger dans le système et utiliser pour leur application, qui soient disponibles immédiatement. La disponibilité dans un système d'une vaste bibliothèque de programmes utilisables de façon commode est probablement un des principaux facteurs pour choisir un système avec succès. En outre, le système doit naturellement offrir les facilités hardware nécessaires et il faut généralement de plus la facilité d'ajouter commodément de nouveaux dispositifs requis pour l'application spécifique que l'utilisateur a en vue.

## Et le microprocesseur XYZ ?

La disponibilité hypothétique d'un nouveau microprocesseur XYZ, deux fois plus rapide, avec deux fois plus de registres etc., n'a aucun intérêt ici. À moins que ce nouveau microprocesseur ne puisse utiliser les bus standards et le logiciel standard qui existent déjà, il n'aura pas un impact notable sur le marché des microordinateurs personnels avant longtemps. Un microprocesseur seul ne constitue pas un système. On a vu que le choix efficace exige la satisfaction simultanée d'un grand nombre de critères différents. La nature du microprocesseur lui-même est de peu d'importance en regard des critères qui ont été présentés.

Etant donnée notre affirmation que les périphériques ont une influence majeure sur le coût et les performances du système, examinons-les maintenant.

# CHAPITRE IX
# LES PERIPHERIQUES

## INTRODUCTION

Le choix des périphériques adéquats peut être plus complexe que celui de l'unité centrale, c'est-à-dire du microordinateur proprement dit. En outre, il a généralement l'influence la plus importante sur le coût et l'utilisation du système.

Nous allons examiner et évaluer dans ce chapitre tous les périphériques usuels : d'abord les unités d'entrée (le clavier), puis les unités de sortie (écran, imprimante), les mémoires de masse (disques, bandes) et enfin les périphériques spéciaux (entrées-sorties vocales, crayons lumineux et autres).

## LE CLAVIER

Le clavier est souvent compris dans le coffret unité centrale, ou encore dans l'unité d'affichage cathodique. Cependant, le plus souvent, les claviers forment une unité séparée.

Il y a relativement peu de choix dans les claviers et c'est heureux. Les critères de choix pour un clavier sont du domaine du bon sens. Le clavier doit être robuste, fiable et muni de bons contacts. La disposition des touches doit être pratique pour l'application considérée. Une caractéristique à examiner est la protection contre *les appuis multiples de touches* (« rollover »). Le problème se présente quand on appuie simultanément, ou presque, sur plusieurs touches. Cela a lieu, par exemple, en saisie de données rapide. Les claviers simples peuvent ignorer ou rejeter les touches enfoncées en trop. Un clavier « protégé » stocke automatiquement les appuis de clés multiples tant qu'ils ne sont pas simultanés.

**Pour l'utilisation en gestion :** un clavier simplement alphanumérique ne suffit pas. Il faut un clavier commercial complet avec :

1. les touches alphanumériques habituelles.

2. un clavier numérique séparé avec les 10 chiffres, plus le contrôle de curseur sur l'écran (haut, bas, gauche, droite) plus des touches de commande (remise à zéro, etc.).

## PERIPHERIQUES DE SORTIE

### L'unité d'affichage à écran cathodique

L'écran est devenu le support de sortie universel pour les systèmes microordinateurs. C'est parce qu'il est silencieux et permet de visualiser une grande quantité de texte en peu de temps en le déroulant verticalement (cela s'appelle le *déroulement*). En outre, un moniteur à écran peut s'acheter pour moins de 800 - 1 000 F ce qui permet de l'intégrer facilement dans un produit.

On peut distinguer trois types d'unités d'affichage à écran.

1. Le téléviseur traditionnel.
2. Le moniteur vidéo simple.
3. Le terminal à écran.

Examinons-les maintenant.

### Le moniteur vidéo

Un moniteur vidéo est simplement un poste de télévision dépourvu de l'électronique d'accord, de séparation et d'amplification ainsi que du haut-parleur, des sélecteurs de canaux, etc. Comme il n'a pas besoin de détecter des fréquences VHF/UHF, le moniteur a uneentrée vidéo directe et il peut accepter des signaux de bande passante bien plus large que dans le cas d'une TV normale. En particulier, on peut avoir des lignes de 80 caractères.

Tout téléviseur contient un moniteur et peut être modifié pour avoir une entrée vidéo directe. Toutefois, ceci est le sujet d'un autre livre (C5).

### Le terminal à écran traditionnel

Le terminal à écran « CRT » est formé d'un moniteur, d'un clavier et de l'électronique de commande (contrôle du curseur, du clavier, des fonctions de l'écran, et mémoire locale). Cela a été le terminal traditionnel pour communiquer avec les ordinateurs et c'est l'un des mieux adaptés au contexte de gestion grâce à sa conception

professionnelle et à ses caractéristiques. Toutefois, c'est la solution la plus chère (5 000 F et plus, selon les options).

La combinaison d'un clavier et d'un moniteur vidéo plus du software peut rendre les mêmes services.

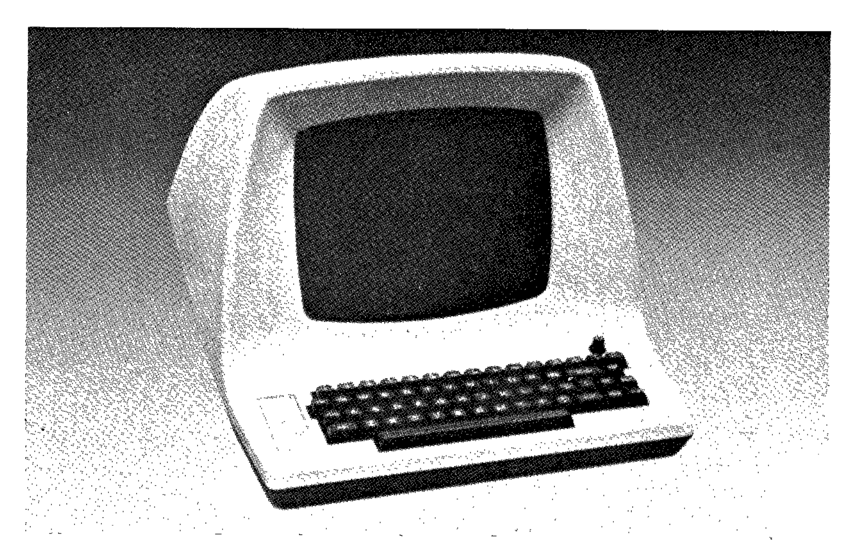


Fig. 9.1 : Terminal « simple » LearSiegler. C'est un terminal à écran, de base. Il a un écran 12 pouces, 59 pouces, 24 lignes de 80 caractères, les connecteurs RS232 et boucle 20mA (64 caractères ASCII, matrice de points 5 × 7).

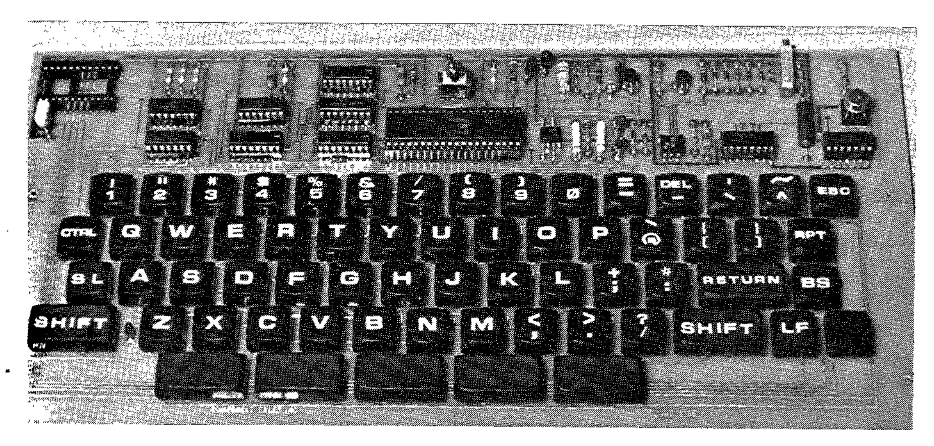



Fig. 9.2 : Clavier QWERTY typique avec son électronique de décodage.

## Le poste de télévision ordinaire

Pour autant qu'on puisse le considérer comme « gratuit » (on l'a déjà à la maison), c'est le terminal le moins cher. Mais si on se connecte sur la prise d'antenne (comme pour les jeux télé) la limitation de la bande passante qui en résulte entraîne une limitation à des lignes courtes sur l'écran.

Cependant, pour avoir un affichage couleur, les autres solutions sont si chères que le poste de télévision est le meilleur choix. Il faut se rappeler que la limitation du nombre de caractères par ligne joue toujours.

Vu le bas prix des moniteurs noir et blanc (et leur bande passante plus large) il n'y a virtuellement aucun avantage à utiliser un téléviseur noir et blanc.



Fig. 9.3: Le Soroc 1Q 120 est un terminal à écran de base, en compétition avec celui vu ci-dessus. Il affaiche 24 lignes de 80 caractères. Il possède un clavier numérique, et a les lettre majuscules et minuscules.

## Affichage de texte sur un écran

Examinons maintenant les caractéristiques communes à toutes les unités à écran. Leur but est d'afficher du texte et, si possible, de permettre un peu de graphiques.

Dans un téléviseur, les images sont formées en éclairant des points minuscules sur l'écran. Pour afficher des caractères, les points sont simplement bien plus gros afin d'avoir des caractères nettement visibles. Chaque caractère est défini par une *matrice de points* dont un exemple apparaît sur la Fig. 9.4. Chaque caractère est formé en éclairant des points dans un rectangle de 7 lignes de 5 points chacune. Pour une meilleure définition, plus de points peuvent être utilisés, par exemple une matrice 7 × 9. Mais cela réduit le nombre de caractères de taille donnée qu'on peut afficher sur l'écran. Dans les systèmes bon marché, on fait appel à la matrice 5 × 7. Il est difficile de distinguer les majuscules et les minuscules et de permettre aux minuscules de se prolonger en-dessous de la ligne avec si peu de points.

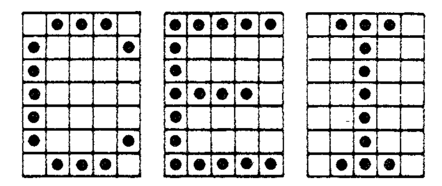


Fig. 9.4 : Matrice de points 5 × 7.

Une question technique intéressante est la suivante : on a montré que le terminal reçoit les caractères de l'ordinateur sous forme d'octets c'est-à-dire de groupes de 8 bits (chaque caractère correspond à une combinaison unique de 8 bits en code ASCII).

**La question est :** *Comment ces 8 bits sont-ils convertis en 35 points (ou plus) sur l'écran ?*

**Réponse :** *Simplement à l'aide de mémoires ROM. Pour chaque code 8 bits, un ensemble d'emplacements mémoire fournit le motif de points à éclairer.*

**Conséquence :** Comme il est simple d'échanger les boîtiers ROM, il est facile de modifier le jeu de caractères. Les écrans professionnels peuvent offrir différents jeux de caractères allant d'APL aux alphabets étrangers.

### Combien de lignes et de caractères ?

Evidemment, l'utilisateur aimerait avoir autant de caractères et autant de lignes que possible. Dans le cas d'un téléviseur, avec l'ordinateur relié à la prise d'antenne, la bande passante limitée

nous restreint à environ 24 lignes de 40 caractères si on a un téléviseur de bonne qualité, avec une matrice 5 × 7 et l'affichage des seules majuscules. Une largeur de seulement 24 caractères par ligne est souvent utilisée.

Dans le cas d'un moniteur ou d'un terminal, l'ordinateur se connecte directement à l'entrée vidéo ce qui améliore considérablement la bande passante. 24 lignes de 80 caractères (majuscules et minuscules) sont pratiquement le standard.

**En gestion :** le nombre de caractères par ligne standard pour les applications de gestion est 80 (largeur d'une carte IBM) et l'idéal serait 120 ou 132. Un téléviseur n'est acceptable que s'il est modifié pour avoir l'entrée vidéo directe, ce qui revient à l'utiliser comme un moniteur. Pour les applications commerciales, en outre, on considère que le nombre de lignes minimum est 24.

Les imprimantes standard impriment 80 caractères par ligne ou même plus (132), d'où l'intérêt évident d'avoir 80 caractères (ou plus) par ligne.

### Possibilités supplémentaires des écrans

La plupart des possibilités supplémentaires sont réalisées par programme, ou, quelquefois, grâce à une carte contrôleur d'écran plus complexe.

— Le curseur : tout terminal à écran doit afficher un curseur indiquant (généralement) la position sur l'écran où le prochain caractère apparaîtra. L'aspect du curseur est typiquement un carré ou un « souligné ». Son aspect peut être programmable et il peut être clignotant.

— Le système peut permettre deux intensités lumineuses (gris et blanc) utiles pour distinguer les questions et les réponses, ou alors on peut inverser le contraste (blanc sur noir).

— On peut faire clignoter les caractères et le curseur.

— Il est possible de disposer de possibilités graphiques limitées. Elles sont réalisées généralement en sélectionnant certains points particuliers dans la matrice de points des caractères.

— La couleur constitue une option évidente. Si l'on utilise un téléviseur couleur, on a généralement quatre « couleurs » : noir, blanc, violet et vert.

### Terminal ordinaire ou intelligent ?

Des possibilités supplémentaires peuvent être obtenues en transférant dans le terminal certaines fonctions qui étaient effectuées par le microordinateur. Le terminal « ordinaire » devient alors terminal « intelligent ».

Un terminal « intelligent » est conçu pour pouvoir servir « off-line » c'est-à-dire déconnecté de l'ordinateur.

Il offre une mémoire locale pour stocker des données, et des fonctions d'édition (pour corriger les erreurs). Vu le bas prix des microordinateurs, on n'utilise généralement pas de terminaux intelligents dans leur cas.

### Une récapitulation des écrans

*1. Pour l'utilisateur individuel :* pour avoir la couleur, la seule option bon marché est d'utiliser le téléviseur couleur familial. Pour du noir et blanc, un téléviseur ordinaire impose une limitation à environ 16 lignes de 32 caractères. Un *moniteur* est recommandé pour une meilleure définition.

*2. Pour la gestion :* quand le coffret microordinateur ne comprend pas de clavier, il faut un *terminal.* Un terminal « inintelligent » est généralement suffisant. Vous aurez 24 lignes de 80 caractères, majuscules et minuscules. Veillez à avoir un clavier complet sur le terminal. On considère qu'un écran de 30 cm est l'optimum. Une matrice de points 7 × 9 ou mieux est nécessaire pour une lecture aisée.

Si le coffret microordinateur comprend aussi le clavier voulu, on peut simplement ajouter un moniteur 30 cm plus la carte contrôleur d'écran nécessaire (si disponible) à l'intérieur du microordinateur et obtenir un fonctionnement équivalent.

### Communication avec l'écran

Plusieurs unités d'entrée ont été mises au point pour pointer vers des emplacements déterminés de l'écran. Les plus fréquemment utilisées seront décrites à la fin de ce chapitre. Ce sont : le crayon lumineux, le manche à balai, et, dans une moindre mesure, la « souris ».

## L'imprimante

L'écran fournit une « copie molle » (temporaire, par opposition à l'imprimante), mais peut déplacer les informations rapidement de bas en haut de l'écran (« déroulement ») ou même passer d'une page à l'autre (« pagination »). Il est souvent désirable ou indispensable, en gestion, de fournir une copie durable sur papier. Une *imprimante* est nécessaire dans presque tous les cas.

*Pourquoi pas tout simplement la Sélectric du bureau ?* Convenablement modifiée, elle peut être utilisée. Mais elle est chère, relativement lente (30 cps : caractères par seconde), et risque de ne pas être assez fiable à moins d'être conçue comme un terminal d'ordinateur. Cette option sera revue ci-dessous.

*Pourquoi ne pas photographier l'affichage sur l'écran ?* Cela ne serait pas très pratique vu la petite taille des photos. En revanche il est techniquement faisable de coupler un terminal à écran et le photocopieur du bureau et cela pourrait devenir une solution dans l'avenir.

Considérons maintenant les différentes possibilités. Naturellement nous voulons :

1. Un bas prix.
2. Une bonne qualité d'impression.
3. Une grande vitesse.
4. Une bonne fiabilité.
5. Un fonctionnement silencieux.

Le choix définitif, comme d'habitude, résultera d'un compromis. On peut distinguer quatre types principaux d'imprimantes :

1. Thermique/électrosensible.
2. A ruban.
3. A matrice.
4. Imprimante - ligne.

Examinons-les maintenant.

## Imprimante thermique ou électrosensible

Les imprimantes thermiques utilisent un papier spécial sur lequel on brûle les caractères. L'élément d'impression possède des points ou des segments qui appuient sur le papier et la combinaison de segments convenable est excitée pour brûler le papier afin de former le caractère voulu.

Comme l'élément d'impression est simple et sans pièces mobiles, ces imprimantes sont bon marché et silencieuses. Mais elles offrent une qualité d'impression médiocre (par comparaison à une bonne Sélectric), elles ne peuvent fonctionner à grande vitesse (à cause du temps nécessaire pour brûler les caractères) et elles exigent un papier spécial.

Les imprimantes électrosensibles et la plupart des imprimantes thermiques sont généralement utilisées pour fabriquer les imprimantes les moins chères fournissant un minimum de possibilités (lignes courtes, basse vitesse). Toutefois, il y a des exceptions notables telles que le « Silent Terminal » de Texas Instruments et elles peuvent aller jusqu'à 90 cps.

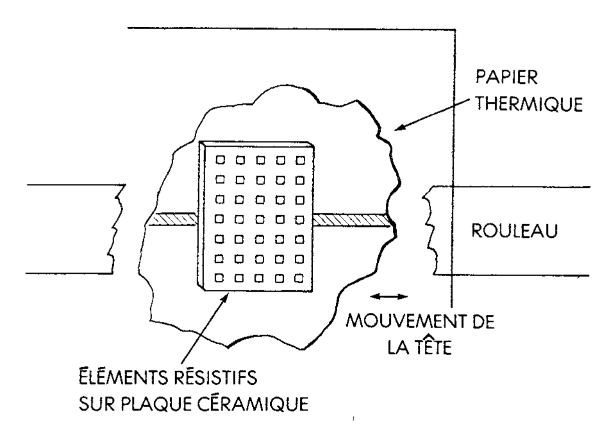



Fig. 9.5: Imprimante thermique à matrice de points.

## Imprimante à ruban

Les imprimantes à *impact* sont les plus fréquemment utilisées. Leur principe de fonctionnement est d'utiliser une tête d'impression qui se déplace, ou alors plusieurs têtes.

Une imprimante à ruban fonctionne essentiellement comme une Selectric, mais en déplaçant la tête devant le papier. Trois techniques principales sont utilisées pour réaliser la tête :

1. Sphère ou cylindre portant tous les caractères.
2. Une marguerite.
3. Une matrice d'aiguilles. Ce dernier cas sera examiné à la section suivante.

Examinons les deux premiers cas.

**La tête cylindrique**

C'est la technique utilisée dans l'un des plus anciens terminaux d'ordinateurs, le Télétype (abréviation de teletypewriter = téléimprimeur ; c'est une marque déposée). Il utilise une tête d'impression cylindrique (cf. fig. 9.6). Le cylindre a 4 rangées de caractères. Il peut tourner sur 17 positions et peut être soulevé. Pour imprimer un caractère, le cylindre tourne jusqu'à la position voulue, monte jusqu'au niveau de la rangée concernée puis est frappé par un marteau. Le caractère appuie le ruban sur le papier. Toutes les liaisons sont électromécaniques à base de cames activées par des électro-aimants.

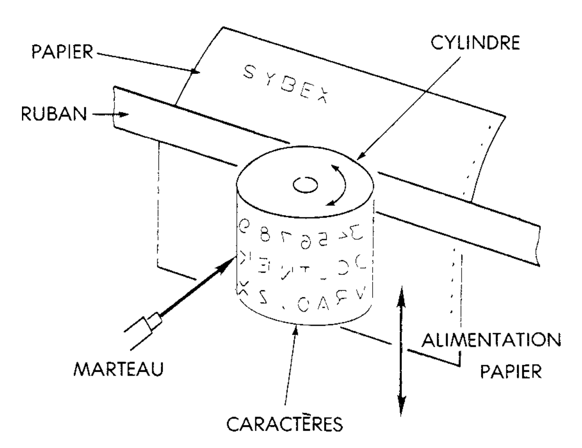


Fig. 9.6 : Le cylindre d'impression du télétype.

Le télétype standard KSR33 fonctionne à 10 cps. Il est bruyant, lent, mais robuste, bon marché (de 7 000 à 10 000 F) et il procure une excellente lisibilité. Il est limité aux ules. Dans sa gamme de prix, il représente un bon choix d'imprimante lente, lisible et fiable. Mais il est inacceptable en gestion.

Il existe beaucoup d'autres modèles. En particulier l'ASR 33 (« ASR » veut dire « émission-réception automatique » tandis que « KSR » veut dire « émission-réception au clavier ») est un KSR 33 muni d'un lecteur-perforateur de ruban pour un coût supplémentaire faible. C'était un moyen valable de conserver des programmes quand les disques n'existaient pas, et cela peut encore servir dans un petit système quand un ASR 33 est disponible. En pratique, cette approche est maintenant périmée avec les systèmes à disques.

Les deux interfaces standard de télétype sont le RS232 C et la boucle de courant 20 mA. Le télétype utilise le code ASCII.

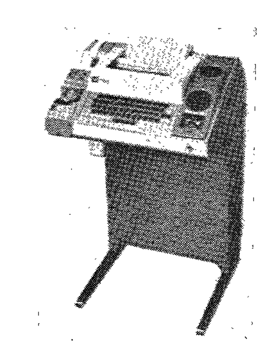


Fig. 9.7 : Télétype traditionnel

**La Selectric**

La Selectric est une machine à écrire IBM transformée. Elle utilise une sphère de plastique métallisé qui peut tourner et basculer. Elle est lente, elle aussi, (15 à 30 cps) mais offre une qualité d'impression identique à celle d'une machine à écrire de bureau ! Son coût est quelque peu plus élevé que le télétype, mais elle peut servir dans une application de gestion minimale. Elle a les majuscules et les minuscules et la sphère d'impression est interchangeable. Mais elle ne convient toujours pas pour un fonctionnement continu ou de longs listings. Continuons notre recherche.

**L'imprimante à marguerite**

L'imprimante à marguerite possède une roue de 100 caractères (voir Fig. 9.8). Chaque caractère est sur un secteur (ou pétale) de sorte qu'on n'a besoin que d'une rotation. Le marteau frappe une seule lettre. En outre, il est possible de garantir un positionnement exact et une frappe uniforme (la pression du marteau est ajustée pour chaque lettre).

Ceci autorise un fonctionnement rapide, silencieux et fiable avec une qualité d'ivpression superbe.

La photo d'une marguerite réelle apparaît Fig. 9.8. De plus, les roues sont interchangeables de sorte qu'on peut utiliser différents jeux de caractères standards chez plusieurs constructeurs (Diablo, Qume, Hewlett-Packard).

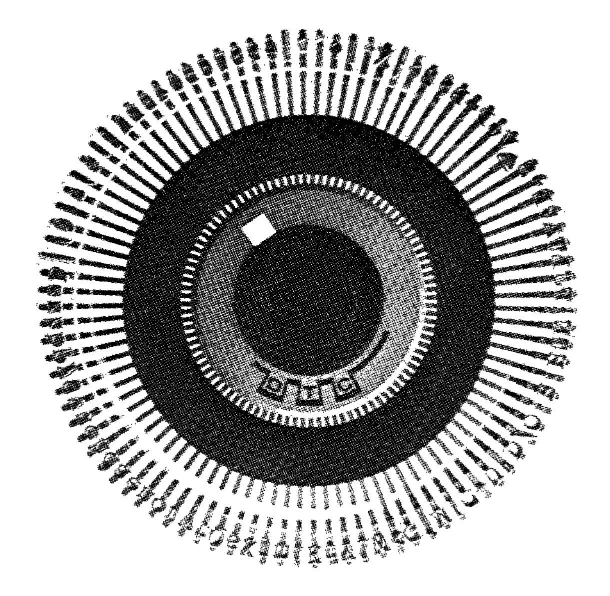


Fig. 9.8 : Marguerite montrant les différents caractères.

ılité d'impression est si élevée que l'on peut souvent of
ıon du texte. Il en résulte des pages prêtes au tirage.

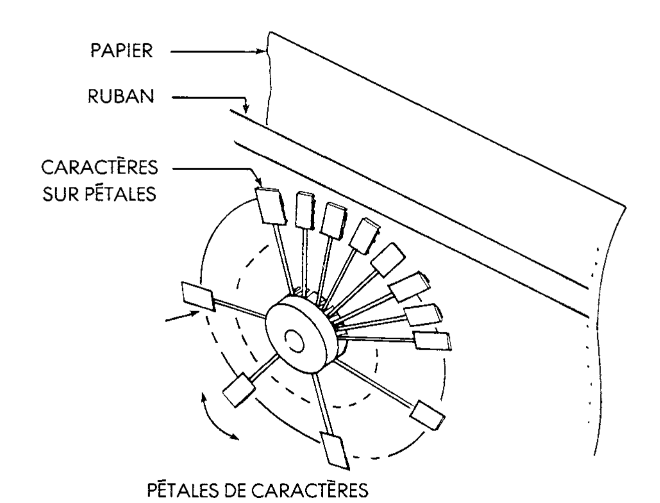



Fig. 9.9 : Rotation de la marguerite.

*Pour l'utilisateur en gestion :* L'imprimante à marguerite est actuellement la solution optimale pour un système microordinateur, mais elle coûte de 20 à 30 000 F, probablement le module le plus cher du système.

**Imprimante matricielle**

L'imprimante matricielle utilise des aiguilles pour imprimer un motif de points sur le papier. Les caractères sont générés comme dans le cas de l'écran, en choisissant la combinaison d'aiguilles voulue dans un rectangle de 5 lignes sur 7 colonnes ou 7 lignes sur 9 colonnes. Chaque aiguille est commandée par un solénoïde. Quand celui-ci est excité, l'aiguille est lancée contre le ruban, marquant un point sur le papier. Pour chaque caractère, il faut imprimer 7 ou 9 lignes de points. Mais l'inertie est minimale de sorte qu'un fonctionnement rapide est possible.



Fig. 9.10 : Le Hytherm de Diablo est une imprimante à marguerite de 94 caractères à 45 cps.

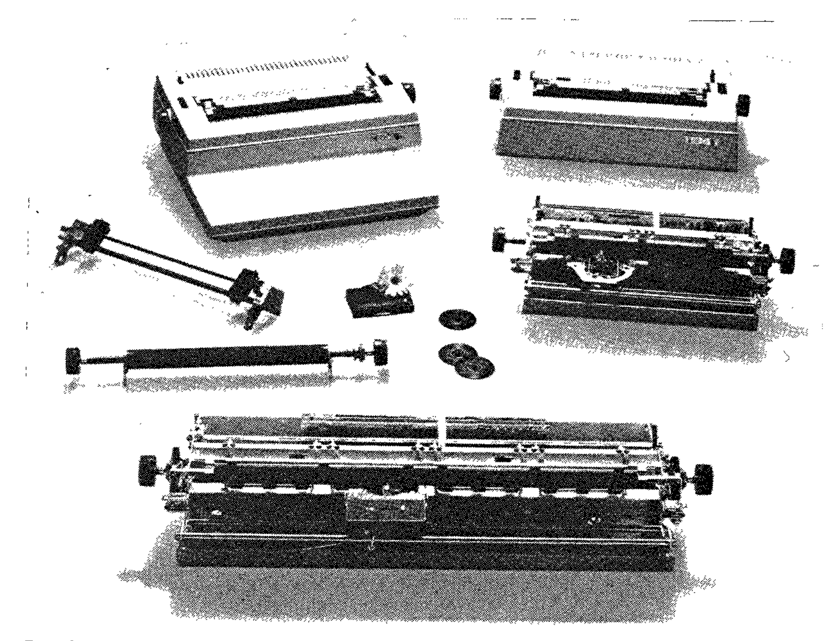


Fig. 9.11 : La famille Sprint Micro 3 de Qume offre la qualité de la marguerite pour un faible coût.

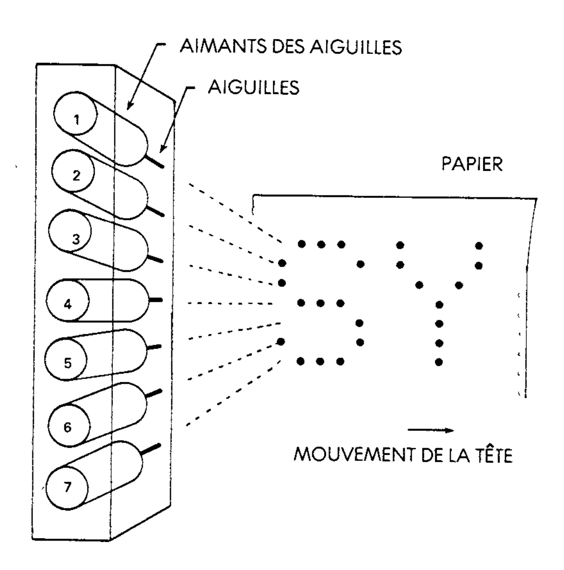



Fig. 9.12 : 7 aiguilles peuvent frapper le papier simultanément.

Dans un système bon marché, on utilise une seule tête qui imprime caractère par caractère et se déplace devant le papier. Dans un système plus cher, toute une ligne d'aiguilles est utilisée pour 80 ou 120 caractères. Toute une ligne est imprimée en 7 ou 9 coups. Ceci est illustré fig. 9.13.

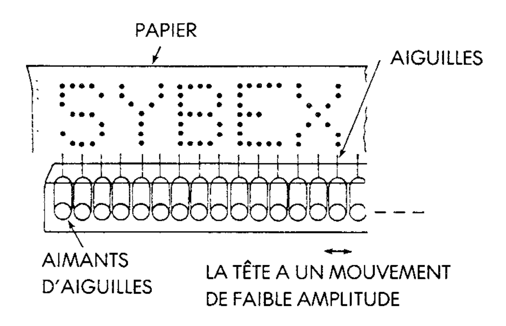



Fig. 9.13 : Toute une ligne de points et imprimés d'un seul coup.

Les avantages généraux sont : coût relativement faible, grande vitesse, silence. L'inconvénient principal est la qualité relativement mauvaise de l'impression. Insistons sur le fait que la qualité suffit à relire le texte, mais ne convient pas à la reproduction ou pour des lettres commerciales.

Une imprimante matricielle connue est par exemple, la Decwriter qui fonctionne à 30 cps et offre une qualité d'impression supérieure à la moyenne.

**Pour l'utilisateur de gestion :** l'imprimante matricielle est une bonne alternative à l'imprimante à marguerite pour un prix moitié moindre, tant qu'on n'a pas à l'utiliser pour taper des lettres commerciales.

**L'imprimante à chaîne**

Une sorte d'imprimante n'a même pas été mentionnée jusqu'ici : l'imprimante à chaîne. Les imprimantes à chaîne peuvent atteindre des vitesses de 1 000 lignes/minute ! Elles sont utilisées avec les gros ordinateurs et quelquefois avec des minis. Mais leur prix est tout simplement si élevé qu'il serait paradoxal de les connecter à un microordinateur bon marché. Si l'utilisateur peut investir la somme élevée que coûte une imprimante à chaîne rapide, il peut aussi investir dans un gros ordinateur pour la piloter.

L'imprimante à chaîne qui apparaît fig. 9.15 est du type à tambour. Le caractère est déplacé autour du tambour jusqu'à la position d'impression voulue.

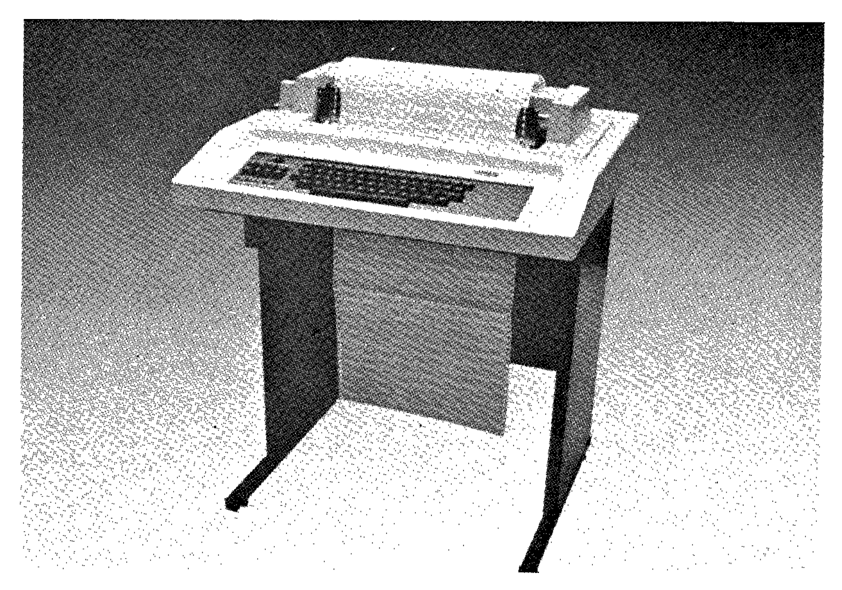


Fig. 9.14: La Decwriter LA36 est une imprimante matricielle à 30 cps de haute qualité. Elle utilise une matrice de poins 7 × 7 et a le jeu complet de 128 caractères:

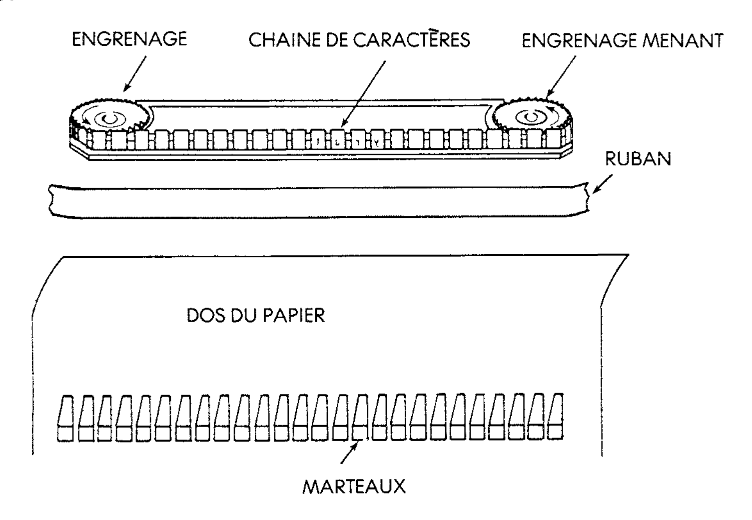



Fig. 9.15: Imprimante à chaine.

**L'imprimante-ligne**

Par opposition aux 200 à 300 lignes/minute maximum des imprimantes précédentes, les imprimantes-ligne vont jusqu'à 3 000 lignes-minute. Nous ne les décrirons pas ici puisque leur prix est hors de proportion pour un système microordinateur.

**Super-imprimantes**

Dans les plus grandes installations informatiques où le coût n'entre plus en ligne de compte, on utilise des imprimantes sans impact pour avoir des vitesses extrêmes. Une imprimante à *jet d'encre* projette de fines goutelettes d'encre qu'elle devie électrostatiquement. On atteint des vitesses de 40 000 lignes-minute.

Dans *l'imprimante à laser,* le papier est chargé électrostatiquement et il attire de la poudre d'encre sèche comme dans une photocopieuse Xerox. Le dessin est ensuite « cuit » dans le papier. On imprime simultanément un grand nombre de lignes et on atteint des vitesses de 20 000 lignes/minute.

**Sommaire des imprimantes pour la gestion**

Trois solutions essentielles ont été indiquées. La meilleure est l'imprimante à marguerite mais elle devient la partie la plus chère du système. Deux autres possibilités sont la Selectric ou une bonne imprimante matricielle.

*Option essentielle :* un entraînement du papier par picots est indispensable si l'on doit imprimer des formulaires commerciaux, des étiquettes ou des chèques. Aucun mécanisme à friction ne peut fournir la précision de positionnement vertical requise.

## DISQUES

Comme la mémoire centrale (RAM) dans le coffret unité centrale, est limitée à environ 48 K octets (le reste étant 16 K de ROM), et est volatile, les gros programmes et fichiers doivent être stockés sur un support permanent. On utilise soit des disques soit des bandes, dont les avantages respectifs vont être présentés. On montrera que le minidisque souple est devenu la meilleure solution pour les amateurs, mais n'est pas adéquat pour la majorité des applications de gestion.

Le disque traditionnel des gros ordinateurs était le disque rigide : c'est un disque revêtu d'oxyde magnétique sur une face ou sur les deux faces et qui tourne en permanence. Une tête de lecture/écriture très semblable à une tête de magnétophone est positionnée au-dessus d'une « piste ». On peut alors « écrire » des données sur le disque sous forme de suite de bits. Les « 0 » et les « 1 » sont inscrits à la surface de la piste en aimantant les particules dans un sens ou dans l'autre. L'information stockée de cette manière est « permanente » tant qu'on n'applique pas au disque un champ magnétique intense.

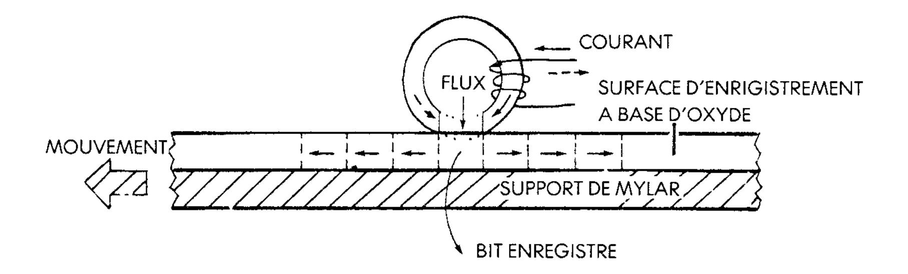



Fig. 9.16 : Enregistrement d'un bit sur disque.

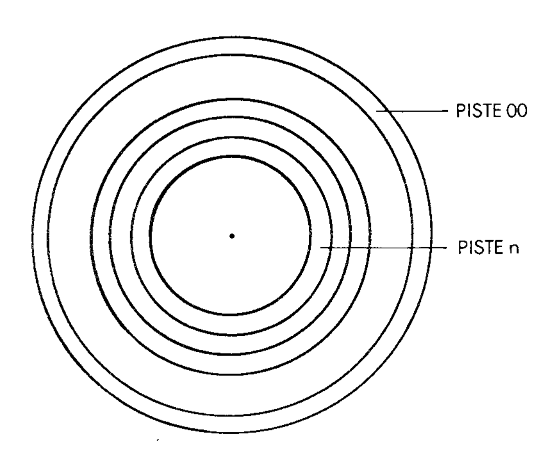



Fig. 9.17 : Les pistes sur le disque.

Les pistes concentriques sur le disque apparaissent fig. 9.17.

Pour pouvoir retrouver l'information sur le disque, il faut savoir sur quelle piste elle est rangée, et aussi où sur la piste.

Pour retrouver commodément l'information sur une piste, il est devenu standard de diviser chaque piste en *secteurs*. Ceci est illustré fig. 9.18. Un secteur typique a 128 octets. L'information est alors retrouvée grâce à son numéro de secteur et son numéro de piste.

Naturellement, personne ne veut se soucier de l'allocation exacte des secteurs à un fichier lors de l'utilisation du système. Un bon DOS (système d'exploitation disque) s'occupe de toutes les utilisations du disque et il incorpore un FMS (système de gestion de fichiers) de façon qu'il soit possible de taper : « LOAD NEW FROM DISQUE », et que aussitôt, les six secteurs attribués au programme « NEW » soient automatiquement retrouvés sur le disque. Cela s'appelle « utilisation de noms de fichiers symboliques ».

Le disque rigide serait un des meilleurs supports de stockage : il permet à la fois grande vitesse et grande capacité. Malheureusement, il est très cher et n'a pas encore été utilisé avec les microordinateurs. Toutefois, il apparaît maintenant de nouveaux disques rigides, plus petits, qui, lorsqu'ils seront au point, pourraient être le meilleur choix pour la gestion (Winchester disk drives).

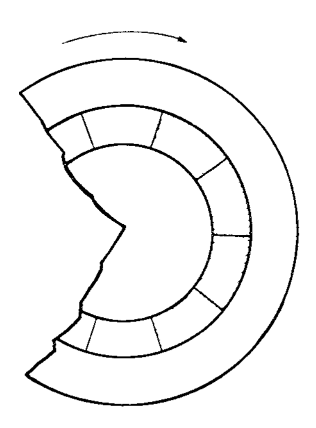


Fig. 9.18 : Un fichier de 4 blocs séquentiels est stocké sur 4 secteurs.




Fig. 9.19 : Disquette et minidisquette.

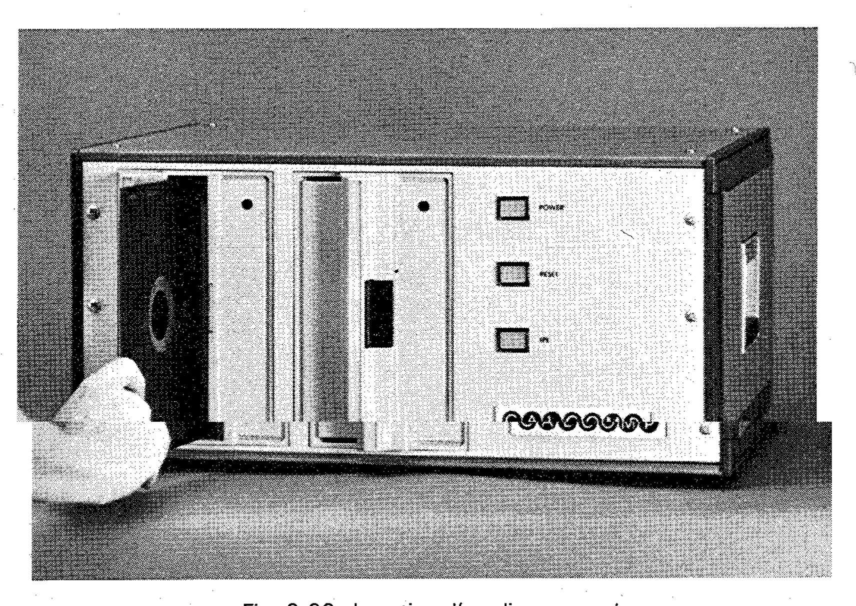


Fig. 9.20 : Insertion d'un disque souple.

## Le disque souple (ou « disquette » ou « diskette »)

Le disque souple fonctionne exactement comme le disque rigide, à part qu'il est plus petit et flexible : le disque est souple et contenu en permanence dans un étui en carton. Il tourne à l'intérieur de l'étui qui a un revêtement à faible frottement. Une ouverture dans l'étui permet à la tête de lecture/écriture d'entrer en contact avec le disque. Pour plus de détails sur le fonctionnement du disque souple, le lecteur peut se reporter aux références C4 et C5.

Les formats d'écriture sur disque souples ont, contrairement aux disques rigides, été standardisés (par IBM) et les prix tendent à être équivalents chez la plupart des constructeurs.

A l'origine le disque souple de 8 pouces (20 cm) fut introduit par IBM (modèle 3740) à la fin des années 60.

Une unité de disque souple est un appareil électromécanique complexe ; elle nécessite une carte-contrôleur qui décode les commandes du microordinateur, leur obéit et gère les transferts de données. Cette carte est normalement ajoutée dans le coffret unité centrale.

## La minidisquette

La minidisquette est une version plus petite du disque souple ; 5,25 pouces (13 cm) de diamètre contre 8 pouces. Elle a été introduite par Shugart en 76 et c'est Shugart qui a établi les formats des données (pas IBM pour une fois). La minidisquette a obtenu un grand succès dans l'industrie de la microinformatique parce qu'elle offre une mémoire de masse à bas prix (la disquette elle-même coûte 25 à 50 F). La minidisquette traditionnelle a 35 ou 40 pistes selon le mécanisme de positionnement de la tête. Un fabricant (Micropolis) fournit même 77 pistes (d'où une capacité doublée) et prétend tout de même à une fiabilité satisfaisante.

La minidisquette normale a en gros 1/3 de la capacité du disque souple habituel. Elle peut être utilisée très avantageusement dans un contexte individuel pour maintenir des fichiers et des programmes personnels.

**En gestion :** La capacité de la minidisquette est trop faible pour la plupart des applications. Le minimum est la disquette normale et encore celle-ci peut être insuffisante pour un gros inventaire ou une grande liste d'adresses.

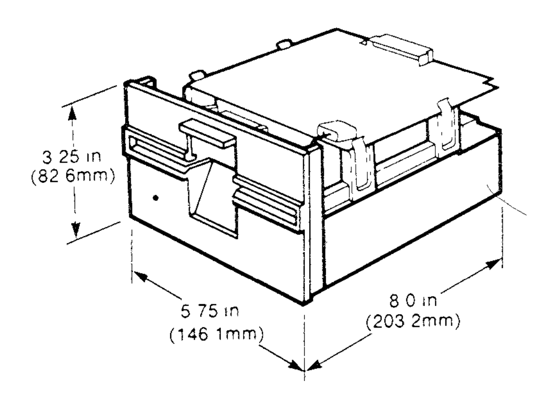



Fig. 9.21 : Dimensions d'un mini-disque souple.

## Augmentation de la capacité

Deux options sont possibles pour doubler la capacité disponible : *double densité* ou *disque double face*. La *double-densité* compacte deux fois plus de données mais perd un peu de fiabilité.

Les unités *double-face* utilisent les deux faces du disque, mais introduisent des problèmes mécaniques.

Toutefois, ces deux techniques murissent rapidement et on peut déjà les utiliser avec assez de sûreté.

## Un ou deux disques ?

Si l'on a l'intention de copier un programme ou un fichier d'un disque sur un autre, il faut *deux* disques. Si l'on a besoin de faire un classement par ordre alphanumérique, ou de fusionner deux disquettes, il faut deux unités. Si l'on a besoin d'une plus grande capacité en ligne, il faut deux, trois ou quatre disques.

Tout système devrait avoir deux disques. Un disque unique limite les capacités du système mais peut être acceptable pour des traitements individuels simples.

**En gestion :** *deux disques forment le minimum indispensable.*

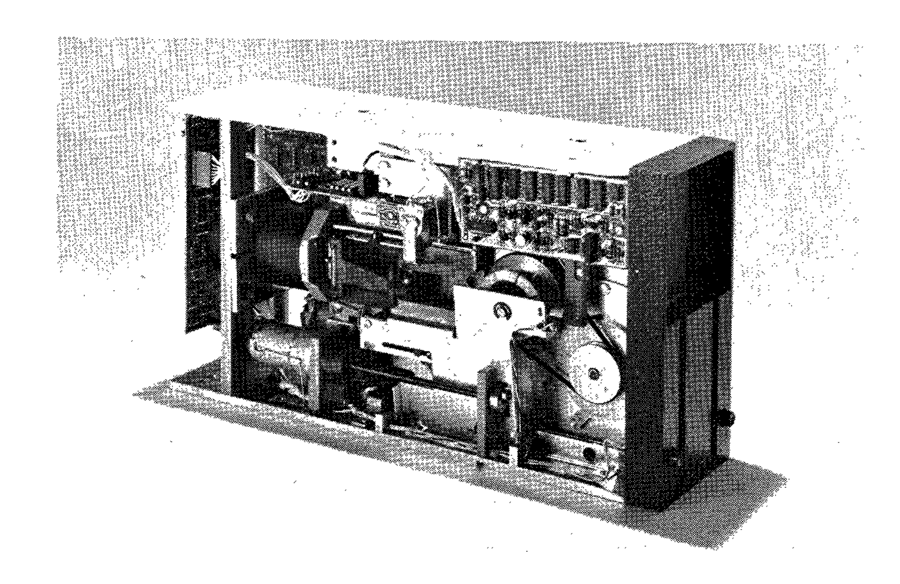


Fig. 9.22: L'unité double Persci 277 gère deux disquettes dans un volume de 8,6 × 4,4 × 15,0 pouces seulement.

### Récapitulation du disque

Le disque souple (ou le mini disque souple) est devenu le système standard de mémoire de masse pour les microordinateurs. Une unité double est nécessaire pour un fonctionnement efficace. La minidisquette est adéquate pour les calculs personnels. Le disque souple normal est un minimum pour la gestion et il pourrait être supplanté prochainement par les minidisques rigides récemment introduits.

## BANDES MAGNETIQUES

Un magnétophone à cassettes du commerce peut être utilisé comme mémoire de masse de faible coût. *Précaution :* un magnétophone standard exige un connecteur de télécommande marche-arrêt et il doit être débranché pour les rebobinages et l'avance rapide. Enfin, le magnétophone à cassettes ordinaires sacrifie le rapport signal sur bruit à la distorsion harmonique (et il ignore la distorsion de phase) ce qui conduit à des problèmes de fiabilité si l'on veut enregistrer en forte densité.

Il existe aussi des unités de cassettes digitales. Elles ont toujours

l'inconvénient d'un long temps d'accès et de leur nature séquentielle (accès direct difficile).

## MEMOIRES DE MASSE DE L'AVENIR

Deux nouvelles technologies ont été développées pour les mémoires, dont le but est de fournir une grande capacité (disons 1 megaoctet = 1 million d'octets), à grande vitesse et bas prix. Ce sont les CCD (systèmes à couplage de charges) et les mémoires à bulles.

Le CCD pourrait remplacer les disquettes dans quelques années, mais il est encore volatile. La mémoire à bulles peut fournir une mémoire permanente très vaste dans un avenir plus lointain, mais elle est encore chère.

## AUTRES PERIPHERIQUES

### Le crayon lumineux

Le crayon lumineux est un périphérique d'entrée-sortie en liaison avec un écran. C'est un moyen de communication puissant et commode permettant de repérer un point déterminé sur l'écran. Il fonctionne tout simplement en décelant quand le point qu'il est censé repérer se trouve éclairé par le faisceau qui balaye continuellement l'écran. Le moment où la lumière est décelée permet à l'ordinateur (ou, plutôt, à un programme) de calculer les coordonnées approximatives du point sur l'écran.

Le crayon lumineux convient particulièrement pour les choix dans un « menu » : il suffit de pointer sur le choix désiré.

Toutefois, il n'est pas très précis et il est cher ; il est donc rarement utilisé avec les microordinateurs.

**Dans un contexte de gestion :** le crayon lumineux n'est probablement pas commode, l'opérateur communique efficacement avec le système par l'entremise du clavier qui lui permet de *confirmer* chaque choix.

Si l'opérateur a accidentellement pointé sur un choix qui n'est pas le bon, il faut reprendre toute la procédure de choix pour le corriger. Avec un clavier, le programme attend un « retour chariot » avant d'exécuter ce qui donne du temps pour vérifier ou corriger.

**Dans un contexte d'enseignement :** le crayon lumineux est idéal, car il dispense les utilisateurs inexpérimentés d'avoir recours au clavier et les erreurs ne sont pas si graves. Un enfant, par exemple, peut pointer sur la bonne réponse sans rien savoir sur les claviers.

**Le manche à balai**

C'est un levier vertical qui peut être déplacé vers la gauche, vers la droite, en avant, en arrière, ou dans toute direction intermédiaire. Il est idéal pour déplacer rapidement un point sur l'écran. Il est très utilisé pour les jeux vidéo et est de très bas prix (et de faible précision).

**La « souris »**

C'est une version plus précise du manche-à-balai. C'est essentiellement un petit appareil muni de roues qui peut être déplacé à la main dans toutes les directions sur une tablette. Le mouvement des roues peut être mesuré avec précision et un point est déplacé sur l'écran en accord avec les déplacements de la souris. Son nom vient du surnom familier qu'on lui avait donné dans les années 60 en raison de sa taille et de son type de mouvement.

Elle est chère et n'est généralement pas utilisée avec les microordinateurs.

**Tablette (RAND)**

Il existe de nombreux types de tablettes à digitaliser sur lesquelles on peut écrire ou déplacer un « stylo » spécial. La position du stylo est détectée et corrélée avec un emplacement sur l'écran. Une bonne résolution entraîne un coût élevé d'où il s'ensuit que les tablettes sont rarement utilisées avec les microordinateurs.

**Entrée vocale**

Oui, il est possible de commander un ordinateur par la parole, à l'aide d'un analyseur vocal convenable et de son programme. Dans la gamme de prix compatible avec les microordinateurs, un tel système accepte un vocabulaire limité (disons une ou deux douzaines de commandes) de courts ordres bien articulés, après une courte période d'apprentissage pour tout nouvel utilisateur. Le

temps de traitement nécessaire peut entraîner que l'ordre ne soit exécuté que quelques secondes après que l'ordre ait été prononcé.

De telles cartes sont disponibles commercialement et peuvent être ajoutées à un système compatible (généralement S100).

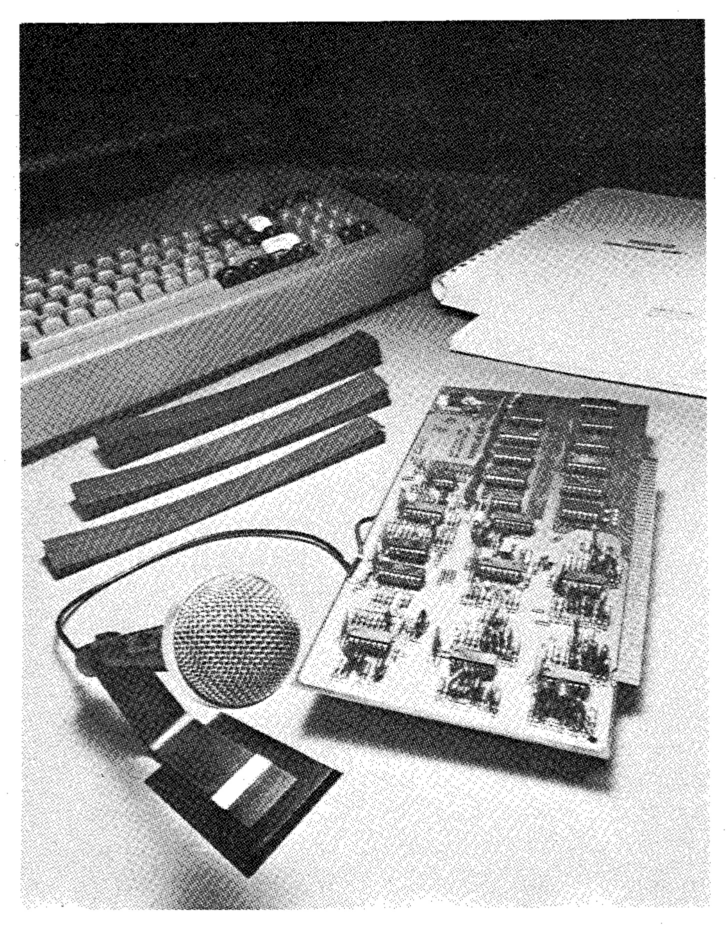


Fig. 9.23 : Le système « Speechlab » de Heuristics permet l'entrée vocale. Il est compatible S100.

## Sortie vocale

La voix peut être synthétisée et un programme peut produire un son vocal acceptable en réponse à un codage spécifique des phonèmes (unités de parole élémentaires). Il nécessite une carte

synthétiseur, un bon programme d'analyse qui code les phonèmes quand on les prononce devant un micro et un gros programme de restitution. Le son de la voix peut être assez bon. Le problème principal réside dans l'analyse précise de la voix pour coder correctement toutes ses particularités : une retouche manuelle peut s'avérer nécessaire.

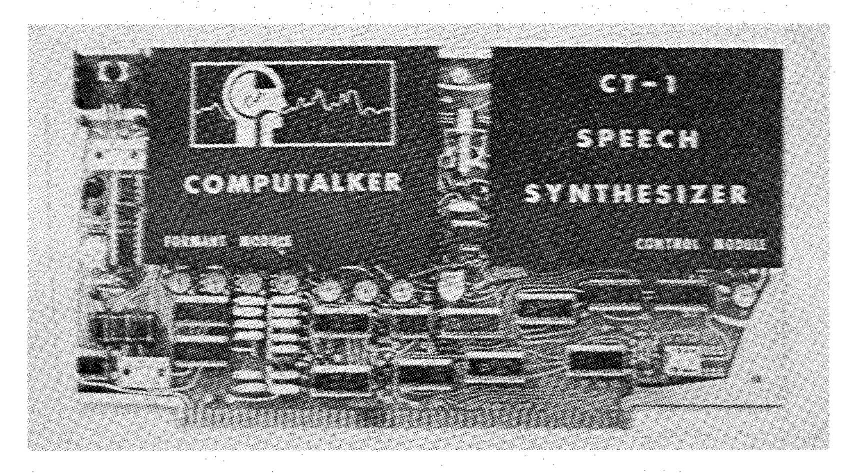



Fig. 9.24 : Le Computalker procure une sortie vocale.

## AUTRES PERIPHERIQUES POUR AMATEURS

### LEDS

Les diodes émettrices de lumière (LED) sont le système d'affichage le moins cher, mais sont limitées aux symboles numériques ou aux symboles hexadécimaux (0 à 9, A à F), si elles doivent rester bon marché. Elles sont utilisées pour les microordinateurs sur une seule carte.

### Interrupteurs

Des interrupteurs de détection peuvent être connectés directement à une carte microordinateur et ils sont utilisés pour réaliser des alarmes anti-vol ou automatiser des réseaux de train électrique, par exemple.

### Relais

Les relais électromécaniques fournissent un isolement convenable entre deux circuits. On peut maintenant monter des relais miniatures

directement sur la carte microordinateur ce qui permet de commander un circuit de puissance considérable, par exemple pour allumer ou éteindre les lumières, commander des moteurs ou d'autres appareils.

### DAC et ADC (convertisseurs analogiques-digitaux)

Si l'on mesure une tension avec un voltmètre digital, on aura par exemple, l'affichage : « 12,5 V ». La tension est une grandeur analogique (c'est-à-dire à variation continue). « 12,5 » est une valeur digitalisée donnant une mesure approximative à 0,1 V de précision près. Un DAC est un convertisseur digital-analogique. Il est nécessaire si l'on veut que l'ordinateur produise un signal analogique qui puisse varier de façon (presque) continue.

Un ADC est un convertisseur analogique-digital. Il était nécessaire dans notre voltmètre. Un ADC est nécessaire pour mesurer n'importe quelle grandeur physique : température, pression, intensité. Il y a maintenant une variété de ADC et de DAC bon marché disponibles en un boîtier.

## RECAPITULATION DES PERIPHERIQUES

Un grand nombre d'options existent pour des besoins spécifiques. Toutefois, pour résumer brièvement :

1. Un système personnel minimum nécessite le microordinateur, un clavier, un moniteur vidéo (ou un téléviseur), un magnétophone à cassettes.
2. Un meilleur système utilise une minidisquette au lieu de la cassette.
3. Un système de gestion exige : un microordinateur avec une grande mémoire interne, un terminal à écran avec un clavier de type commercial, deux disques souples de taille normale et une imprimante avec avancement du papier à picots.

### En conclusion

— Tout système microordinateur a un écran et un clavier.

— Presque tous les systèmes utilisent un magnétophone à cassette ou un disque.

— La plupart des systèmes nécessitent une imprimante.

L'analyse des caractéristiques des périphériques présentés dans ce chapitre devrait fournir un bon guide pratique pour le choix des périphériques ; dans la plupart des systèmes, ils représenteront la part la plus importante du coût total.

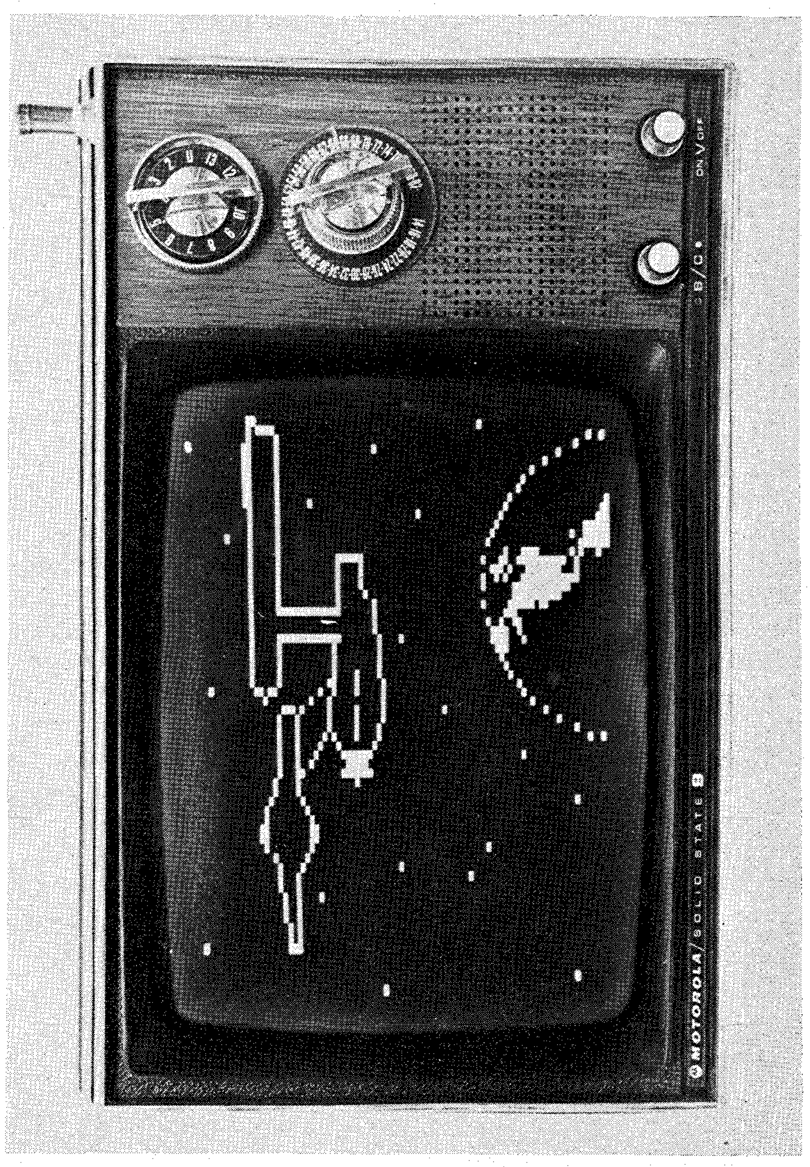
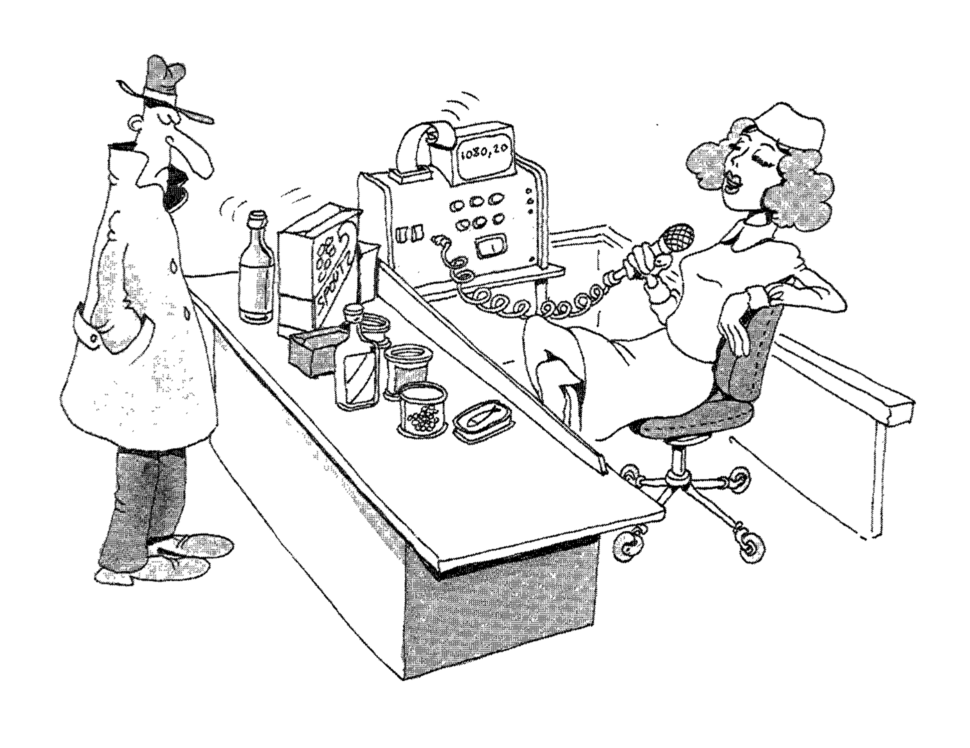



Fig. 9.25 : Le Vaisseau U.S.S. Enterprise dans l'espace. (Un téléviseur n'offre que des possibilités graphiques limitées à cause de l'étroitesse de la bande passante).




L'entrée vocale simplifie désormais les communications

# CHAPITRE X
# CHOIX D'UN MICROORDINATEUR

## MICRO-HISTORIQUE DES ORDINATEURS

Le calcul scientifique appliqué à l'astronomie, la navigation, la géographie, et l'enregistrement des évènements remonte à plusieurs milliers d'années. L'abaque des anciens Chinois est un exemple des premiers essais de mécanisation des opérations de calcul simple.

Le progrès important suivant sur la voie du calcul automatique fut marqué par la machine à additionner mécanique de Pascal (1643, France). Elle fut inventée par le jeune Pascal pour faciliter des calculs de comptabilité, et elle faisait appel à des roues tournantes à engrenages, de sorte qu'une transition de 9 à 0 faisait tourner la roue suivante d'un cran (cela s'appelle maintenant ajouter la retenue).

La prochaine étape conceptuelle a été probablement l'ouvrage de Babbage sur sa « machine analytique » (1820 à 1834, Grande-Bretagne). Regardé généralement comme un excentrique, Babbage ne put faire réaliser sa machine, mais il avait de fait établi le modèle des ordinateurs d'aujourd'hui. Son travail était en avance sur son temps et ne fut pas compris.

L'étape suivante fut l'introduction des premières calculatrices à la fin de la seconde guerre mondiale (1944-45). La MARK I fut développée à Harward par Howard Aiken, avec des fonds d'IBM, tandis que l'ENIAC fut développée par Eckart et Mauchly sur des fonds de l'armée américaine à la Moore School de l'Université de Pennsylvanie.

La MARK I était un « ordinateur de 1re génération », faisant appel à des relais électromécaniques. Elle fonctionna de 1944 à 1959.

L'ENIAC (Electronic Numerical Integrator and Computer = intégrateur numérique et calculateur électronique) était une machine de « 2e génération » utilisant des tubes électroniques (18 000) qui fonctionna de 1946 à 1958. Cette machine géante avait été construite pour calculer des trajectoires de missiles. Les changements de programmes demandaient des jours et ils

nécessitaient un recâblage et de la soudure. L'ordinateur à « programme enregistré » n'était pas encore inventé. L'ENIAC occupait une grande salle de la Moore School of Electrical Engineering, et ne fonctionnait que pendant quelques heures entre deux pannes. L'ENIAC n'avait que 1 K bits de mémoire de travail (contre 400 K ou plus pour un ordinateur contemporain) ; elle avait 10 000 condensateurs, 65 000 résistances, 7 300 relais ou interrupteurs et occupait 3 000 m3. Elle consommait 160 KW.

A ce moment, IBM examina les résultats, décida que les ordinateurs n'avaient pas d'avenir et se retira du marché.

En 1945, Von Neumann avait finalement formulé le concept d'un ordinateur à programme enregistré, où tant le programme que les données seraient stockés dans une mémoire permettant d'exécuter n'importe quel programme en toute généralité. L'ENIAC devait alors être suivie par nombre de ---ACs (Automatic Computers) :

— EDVAC (Electronic Discrete Variable Automatic Computer)
— EDSAC (Electronic Delay Storage Automatic Computer)
— UNIVAC (UNIVersal Automatic Computer) créé en 1951 par Eckart et Mauchly pour le bureau de Recensement Américain. IBM s'aperçut alors qu'ils avaient mal évalué le marché et ils mirent tout leur poids pour s'en emparer. Le lecteur appréciera dans quelle mesure ils y ont réussi.

IBM présente le 701 en 1953 et prend une grande part du marché.

Les transistors deviennent une réalité à la fin des années 50 et ils entraînent l'introduction de la 3[e] génération.

Les années 64-66 sont marquées par des événements importants.

IBM introduit la série 360 qui donne à la compagnie sa domination mondiale.

DEC (Digital Equipment Corporation) introduit le PDP 6 (1) qui sera suivi du PDP8, le miniordinateur qui a eu le plus de succès au monde.

A l'autre extrémité du spectre, CDC (Control Data Corporation) introduit le CDC 6600, superordinateur destiné aux calculs numériques les plus complexes.

---

(1) PDP = Programmable Data Processor, système programmable de traitement de données. IBM avait en effet breveté le terme « ordinateur » (N.D.T.).

**La quatrième génération**

Au début des années 60, la région de Sunnyvale, au sud de la baie de San Francisco aux Etats-Unis, était encore une zone agricole paisible, connue pour ses vergers. Elle est maintenant devenue une zone industrielle intensive, connue sous le nom de « Silicon Valley » (Vallée du Silicium). C'est là que sont établis la majorité des fabricants de semiconducteurs. Au cours des années 60, il devient possible d'intégrer de plus en plus de transistors et d'autres composants sur une seule pastille (« puce ») de silicium, de sorte qu'au début des années 70 on peut réaliser plusieurs milliers de transistors sur une seule puce : ceci ouvrait l'ère de LSI (intégration à grande échelle). Le nombre élevé de transistors permet de réaliser un ordinateur simplifié (ou plutôt son unité centrale, cf. chapitre III) en un seul boîtier.

En 71, le premier « microprocesseur », le 4004, fut annoncé par Intel, en ce temps-là une petite société de Silicon Valley. Il fallut plus d'un an à Intel et aux autres fabricants de semiconducteurs pour se rendre compte qu'un nouveau composant révolutionnaire venait de faire son apparition — par accident. C'est alors que furent introduits la plupart des principaux microprocesseurs que l'on connait aujourd'hui : l'Intel 8080, le Motorola 6800, le Zilog Z80, le MOS Technology 6502, etc.

Toutefois, les applications des microprocesseurs étaient confinées aux domaines de l'informatique, du contrôle industriel ou de l'avionique. Comme pour les ordinateurs à leurs débuts, ils étaient utilisés à des applications « scientifiques » et pas encore au traitement de données. Personne n'avait encore pleinement apprécié leurs capacités potentielles.

L'événement significatif suivant eut lieu en janvier 1975 avec l'annonce par MITS de l'ordinateur personnel ALTAIR. MITS était alors une petite compagnie de l'état du Nouveau Mexique, aux USA. Au cours des années suivantes les sociétés de microordinateurs se répandirent à travers le pays pour s'attaquer à ce nouveau marché. La plupart ne pouvaient pas livrer en quantités suffisantes, et certaines ne le peuvent toujours pas. Ce chapitre présente une revue détaillée des produits disponibles.

## LES TYPES DE MICROORDINATEURS

Les critères applicables pour le choix de l'unité centrale et des

périphériques d'un microordinateur ont été présentés dans les chapitres précédents. Nous pouvons maintenant les appliquer aux systèmes existants pour évaluer leurs avantages et inconvénients.

Jusqu'à présent deux formats de présentation ont été distingués :

1. le microordinateur sur une seule carte
2. le système microordinateur.

Nous pouvons maintenant distinguer un troisième format :

3. le microordinateur intégré.

Dans cette dernière forme, l'unité centrale et le clavier ou l'écran sont intégrés dans le même coffret. C'est la forme utilisée pour les systèmes de faible coût utilisés à la maison. Les avantages sont la facilité de manipulation, un prix du coffret plus bas, une apparence agréable. L'inconvénient est généralement le manque de place à l'intérieur du coffret pour des extensions (mémoire supplémentaire ou cartes d'interface).

Examinons maintenant les trois catégories.

*Avertissement :* Vu le rythme rapide d'introduction de nouveaux systèmes (et de retrait de certains), cette revue ne prétend pas être complète. Les prix peuvent, eux aussi, varier rapidement, ainsi que certaines caractéristiques techniques. Cette analyse est présentée uniquement pour sa valeur éducative. Le lecteur doit se référer à la brochure du constructeur pour les prix et spécifications du moment.

## MICROORDINATEURS SUR UNE SEULE CARTE

Les limitations des microordinateurs sur une carte ont déjà été mises en évidence. En dehors des domaines d'ingéniérie et du contrôle de processus, leur intérêt essentiel réside dans leur valeur éducative pour apprendre la programmation élémentaire et les techniques d'interface. La plupart des fabricants de microprocesseurs fournissent un microordinateur sur une carte, muni d'un clavier minimum et de 4 à 6 afficheurs à LED qui peut être utilisé dans ce but, pour un coût minimal.

Plusieurs microordinateurs sur une carte ont spécialement été conçus comme outils de formation bon marché avec le clavier et les LED sur la carte. Ils méritent d'être brièvement mentionnés ici.

### Le KIM-1

Introduit au départ par MOS Technology (qui est maintenant une division de Commodore), le KIM-1 est maintenant disponible aussi

chez Rockwell International (2nde source du 6502). Un « KIM-1 amélioré » a en outre été introduit sous le nom de SYM par Synertek (autre seconde source du microprocesseur 6502 autour duquel le KIM-1 est bâti).

Le KIM-1 a sur la carte 1 K de RAM, deux boîtiers combinés ROM-RAM-entrée-sorties-temporisateur (d'où 2 K de ROM), un clavier 23 touches, six afficheurs à LED, 15 lignes d'entrées-sorties utilisateur ainsi que les interfaces pour un télétype et un magnétophone à cassettes ordinaire.

Pour utiliser le KIM, on n'a besoin que d'une alimentation. Pour se relier à un magnétophone, on n'a qu'à connecter entrée-audio et sortie-audio. La connexion à un télétype est tout aussi simple.

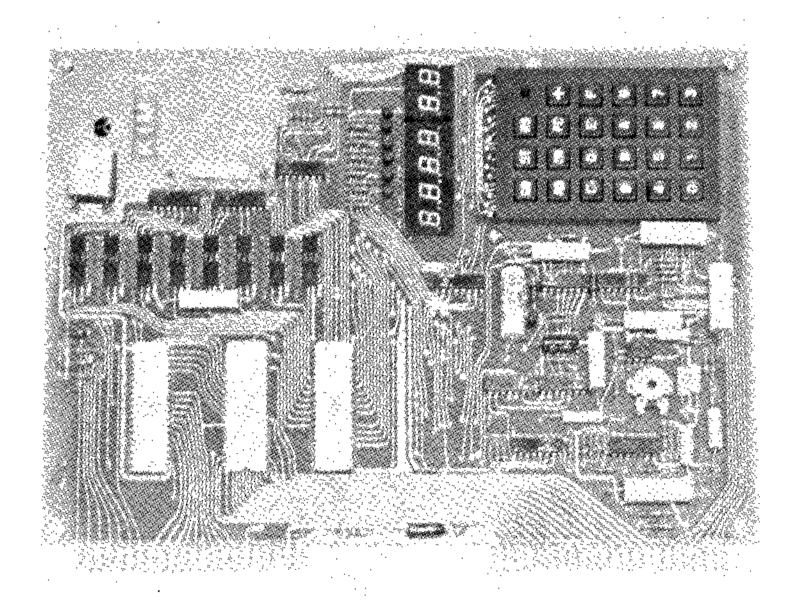


Fig. 10.1 : Le microordinateur sur une carte KIM-1.

### NEC TK80

Microordinateur sur une carte compacte, le TK-80 a été conçu comme outil de formation au 8080A (qui est aussi fabriqué par NEC au Japon). Il est muni d'un clavier hexadécimal avec en plus 9 touches de commandes du moniteur, et de 8 LEDS pour l'affichage.

Une partie de la carte est laissée libre pour des câblages de l'utilisateur.

Les ROM + PROM et la RAM sont extensibles jusqu'à 1 K octets sur la carte. Le moniteur fourni par le constructeur réside en ROM entre les adresses 0000 et 02FF (755 octets). Il est possible d'avoir une alimentation de secours de la RAM par batterie.

## SYSTEMES DOMESTIQUES DE COUT MINIMAL (SYSTEMES INTEGRES)

Les systèmes de coût minimal ont un prix-cible inférieur à 15 000 francs pour un système complet utilisable, et on en trouve pour moins que cela. Nous allons les passer en revue.

### Unitech Videobrain

*Caractéristiques :*

- *Microprocesseur utilisé : Fairchild « F8 » (3870)*
- *Mémoire standard : 4 K de ROM, 1 K de RAM*
- *Clavier : 36 touches, 71 caractères*
- *Entrées-sorties : TV couleur (prise d'antenne), RS232 et boucle de courant.*

Le Videobrain (« cerveau vidéo ») est un ordinateur à clavier intégré conçu comme un développement des jeux télé. Il utilise des cartouches de programmes qui ajoutent jusqu'à 4 K de ROM et il est livré avec des manches à balai, et une prise d'antenne pour se connecter à la télévision couleur qu'on a à la maison.

Il est vendu dans les grands magasins « Macy's » aux Etats-Unis.

Cet ordinateur a été conçu exclusivement pour l'utilisateur qui ne veut pas programmer et qui veut seulement se préoccuper d'enficher un nouveau programme. L'utilisateur dépend totalement du constructeur pour la fourniture de nouveaux programmes et la machine ne peut pas facilement recevoir d'extensions.

Toutefois, vu qu'une majorité d'utilisateurs ne veulent pas apprendre à programmer, il peut trouver un marché important si les

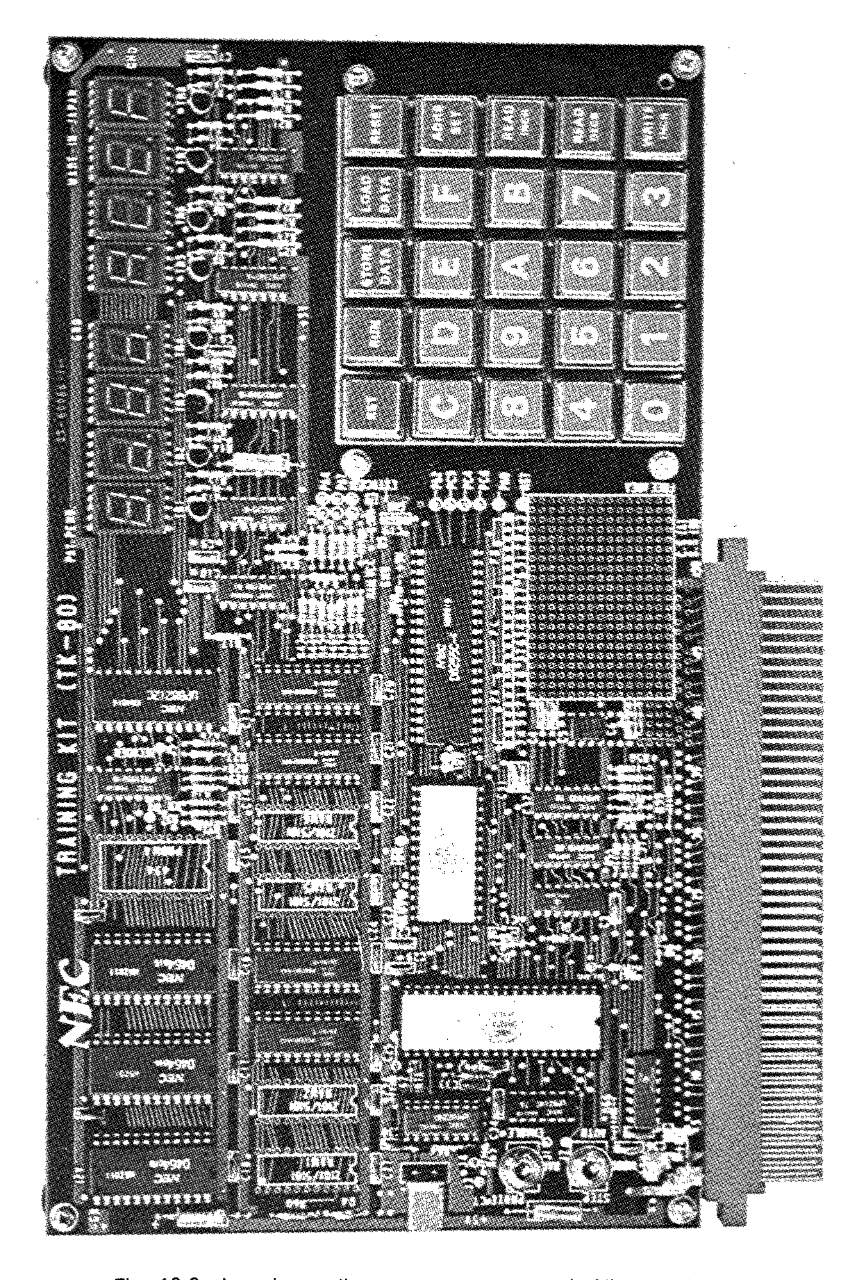



Fig. 10.2 : Le microordinateur sur une carte de Nippon Electric.




Fig. 10.3 : Un « kit » Motorola.

cartouches de programmes arrivent à être disponibles en nombre suffisant.

Utilisations typiques : jeux télé, gestion des dépenses du ménage, calculs de prêts, leçons de maths, leçons de musique, leçons de dactylographie.

**Arcade de Bally**

*Caractéristiques :*

- *Microprocesseur utilisé : Zilog Z80*
- *Mémoire standard : 8 K de ROM, 4 K de RAM*
- *Clavier 24 touches, 70 caractères*
- *Entrées-sorties : sortie prise d'antenne télé couleur. On affiche 12 lignes de 27 caractères.*
- *Mémoire additionnelle : 8 K octets de ROM en cartouches. Extensible à 44 K avec le coffret d'extension contenant 16 K de ROM, 16 K de RAM, un clavier alphanumérique standard et l'interface IEEE488.*

Conçu comme un super jeu télé, ce système possède un clavier à touches mini-BASIC et caractères ordinaires. Les programmes sont stockés dans des cartouches de ROM et l'unité permet de ranger 15 cassettes. Un coffret d'extension a été annoncé : il permet de transformer le système en ordinateur général complet muni d'un clavier alphanumérique standard.

Utilisations typiques : jeux télé, leçons de math, finances du ménage.

**Le P.E.T. de Commodore**

*Caractéristiques :*

- *Microprocesseur utilisé : 6502*
- *Mémoire standard : 8 K de RAM, 14 K de ROM (comprenant l'interpréteur BASIC 8 K)*
- *Clavier 73 touches, 128 caractères*
- *Moniteur télévision : intégré, écran de 9 pouces, noir et blanc, 25 lignes de 40 caractères sur matrice de points 8 × 8*
- *Magnétophone à cassettes intégré*
- *Entrées-sorties : interface standard IEEE488*
- *Avantage spécial : possibilités de traitements semi-graphiques (64 caractères graphiques).*

P.E.T. signifie « Puissant Ensemble de Traitement » (en anglais ordinateur électronique personnel de traitement de transactions. Rappelons que « pet » désigne en anglais un animal familier). Il s'agit d'un système intégré complet avec son propre écran et sa propre mémoire de masse (un magnétophone à cassettes). Les



Fig. 10.4 : Le Videobrain.

programmes peuvent être chargés à partir de cassettes ou, à l'avenir, à partir d'autres périphériques connectés à l'interface IEEE488. Si ce n'est pour la quantité limitée de mémoire RAM interne qui limite la taille des programmes, c'est un ordinateur d'usage général complet. Il est muni d'un clavier alphanumérique complet avec en plus 64 caractères graphiques ; il a les lettres majuscules et minuscules (mais les minuscules et les caractères graphiques s'excluent mutuellement).




Fig. 10.5 : L'Arcade, de Bally, utilise des cartouches-programmes.

La ROM de 14 K comprend l'interpréteur BASIC (8 K), le système d'exploitation (4 K), les routines de diagnostic (1 K), le moniteur permettant d'utiliser le langage machine (1 K). L'interpréteur BASIC est très complet : il autorise l'arithmétique flottante avec 10 chiffres significatifs, les traitements de chaînes de caractères, les tableaux multi-dimensionnés ainsi que l'adressage de cellules-mémoire individuelles (commandes PEEK et POKE).

L'absence de connecteur standard RS232C est peut-être un inconvénient car elle restreint le choix des périphériques facilement connectables. Toutefois, des adapteurs RS232C existent.

Utilisations typiques : jeux, autoapprentissage, finances du ménage, applications de gestion simplifiées.

Fig. 10.6 : Le P.E.T. est un système intégré.

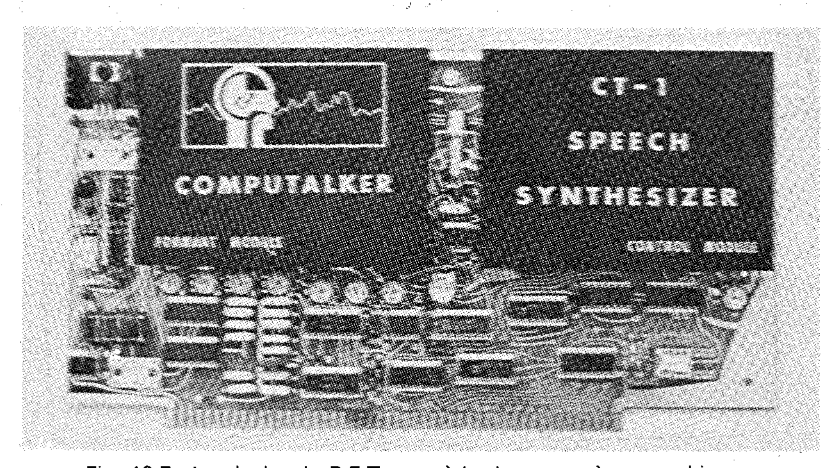



Fig. 10.7 : Le clavier du P.E.T. possède des caractères graphiques.

## Le TRS80 de Radio-Shack

*Caractéristiques :*

- *Microprocesseur utilisé : Z-80*
- *Mémoire standard : 4 K de ROM, 4 K de RAM. Extensible jusqu'à 12 K de ROM et 16 K de RAM à l'intérieur du coffret.*
- *Clavier 53 touches, style machine à écrire.*
- *Ecran : téléviseur 30 cm compris dans le prix. 16 lignes de 64 caractères. Graphiques : résolution 128 (en horizontale) par 48 (en vertical).*
- *Entrées-sorties : magnétophone à cassettes compris dans le prix (250 bits par pouce ; aucune correction d'erreurs). Port d'extension.*

En concurrence directe avec le P.E.T., c'est lui aussi un système complet, avec clavier, moniteur vidéo et magnétophone à cassettes séparés. D'autres périphériques sont disponibles parmi lesquels disque souple et imprimante. Ils doivent être connectés à l'aide du port d'extension spécial au dos du coffret clavier/UC. Le nouveau BASIC niveau II est un BASIC normal et complet, avec possibilité d'utiliser un assembleur pour programmer en langage machine. Les programmes sont normalement chargés à partir des cassettes.

Toutes les commandes sont entrées à partir de son clavier machine à écrire « QWERTY » standard.

Un adaptateur au bus S100 est en projet. Il permettrait la connexion (extérieure) de cartes S100. Actuellement il y a un connecteur spécial 40 contacts à l'arrière du clavier.

## Autres microordinateurs

Le « Sorcerer » de Exidy a été récemment introduit et concurrence directement le Arcade par exemple. Il offre des cartouches de programmes complexes.



Fig. 10.8 : Le TRS-80 est complet mais à éléments séparés.

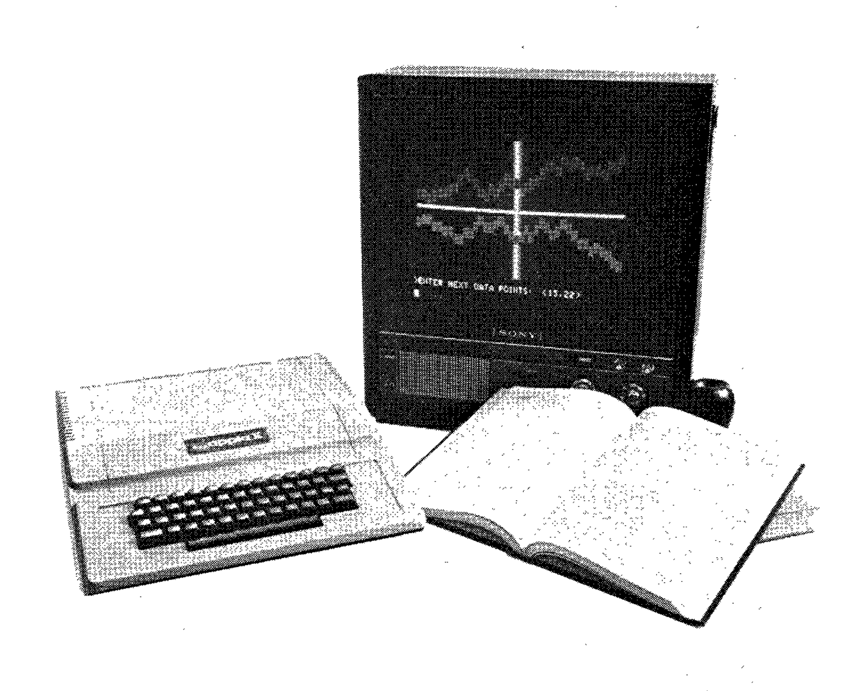


Fig. 10.9 : L'Apple a un clavier intégré et se connecte à un téléviseur couleur.

## APPLE II

*Caractéristiques :*

■ *Microprocesseur utilisé : 6502*

■ *Mémoire standard : 8 K de ROM (y compris BASIC), 4 K de RAM, extensible jusqu'à 48 K de RAM, à l'intérieur du coffret.*

■ *Clavier : 52 touches, style machine à écrire*

■ *Entrées-sorties : sortie sur prise d'antenne de téléviseur couleur (1). 24 lignes de 40 caractères (majuscules en matrice 5 × 7). Possibilités graphiques : résolution 280 (horizontal) par 192 (vertical) en quatre couleurs. Comprend interface cassettes, haut-parleur, connecteurs pour raquettes de jeux vidéo.*

---

(1) système NTSC. Il existe un adaptateur SECAM (N.D.T.).

En apparence, ce système ressemble au Vidéobrain ou à l'Arcade, à cause du clavier intégré. Mais il offre la possibilité d'étendre dans le coffret-même la mémoire à 48 K de RAM (sur la carte) et deux supports pour ROM supplémentaires. Il en résulte une possibilité totale de calcul sans aucune limitation due à la mémoire. Toutefois, seules quelques (petites) cartes d'interface peuvent être ajoutées.

Il nécessite un téléviseur extérieur mais il comprend l'interface couleur ainsi qu'une interface pour magnétophone standard et des interfaces de raquettes.

Les 8 K de ROM comprennent un « moniteur » de 2 K (avec un mini-assembleur, un désassembleur, un programme de dépannage et des routines d'arithmétique flottante) plus un interprêteur BASIC de 6 K avec certaines limitations (entiers seulement, compris entre — 32767 et + 32767, tableaux à une seule dimension). Un nouveau BASIC plus puissant est maintenant annoncé.

Le jeu de caractères ne comprend que les majuscules.

Les programmes sont normalement entrés à partir de cassettes lues sur un magnétophone standard.

Utilisations typiques : jeux télé couleur, calculs d'usage général.

**Altair 8800-b**

*Caractéristiques techniques :*

■ *Microprocesseur utilisé : 8080*

■ *Mémoire standard : quantité quelconque de ROM ou RAM jusqu'à 64 K*

■ *Entrées-sorties : port d'entrée-sortie série en RS232 et boucle de 20 mA.*

■ *Sur option, possède un panneau de commande intégré. On peut mettre 16 cartes S100 supplémentaires dans le coffret de base.*

L'Altair a été historiquement le premier microordinateur présenté et c'est un ordinateur général. Il a créé le bus S100 qui est devenu un standard. Le client peut spécifier quelles cartes incorporer dans le coffret de base. Toute une variété de cartes sont disponibles : mémoires, interfaces, cartes spécialisées.

Pour pouvoir être utilisé, ce coffret microordinateur doit être relié à un terminal à écran (avec clavier), à une mémoire de masse (par ex. disque), et sur option, à une imprimante.

Un software étendu est disponible : DOS, BASIC temps partagé, Assembleur, éditeur, programmes de gestion. Son BASIC version

4.0 est devenu un standard de référence (pas nécessairement le meilleur, mais la référence des BASIC complets).

*Pour la gestion :* Il y a maintenant du logiciel disponible auprès d'une compagnie séparée : Altair Software Distribution Company, 27111 Erwin St., Woodland Hills CA 91367. Les programmes de gestion courants comprennent du traitement de textes, des inventaires, paies, effets à recevoir, effets à payer.

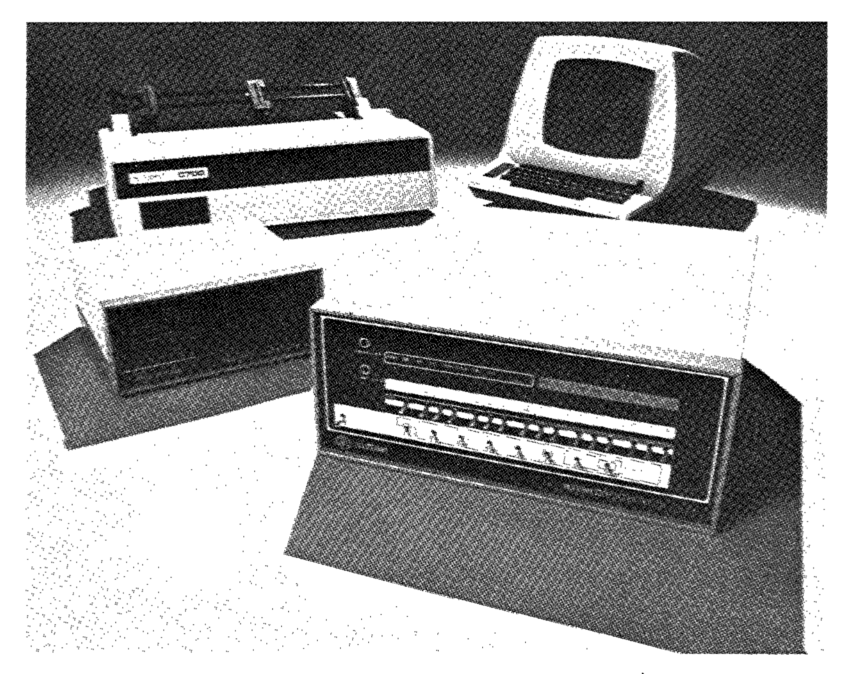


Fig. 10.10 : Systèmes Altair dans l'une de ses multiples versions.

**Imsai 8080**

*Caractéristiques :*

- *Microprocesseur utilisé : 8080 A*
- *Mémoire : toute quantité de ROM ou RAM jusqu'à 64 K.*
- *Entrées-sorties : dépendent des cartes interfaces choisies.*
- *Particularités : cartes fond de panier S100 (22 connecteurs). Panneau de contrôle.*

L'Imsai est un concurrent direct de l'Altair : il a été introduit peu de temps après lui. Il fait appel au bus S100 de sorte qu'il peut, lui aussi, accepter tout un choix de cartes S100.

Les caractéristiques sont, pour l'essentiel, analogues à celles de l'Altair.

Il a, naturellement, un interpréteur BASIC et il peut exécuter le software du 8080 : tout programme écrit en code 8080 tourne aussi sur l'IMSAI.

*Pour la gestion :*
une variété de logiciels indépendants ont été écrits pour l'IMSAI ou pour l'ALTAIR, (ou autres ordinateurs compatibles S100). Ils ne sont pas compatibles s'ils sont écrits en BASIC. Vous devez vérifier dans quel BASIC ils ont été écrits, c'est-à-dire, avec quel interpréteur ils fonctionnent. Il est vrai de dire que tout programme BASIC peut être adapté à un autre interpréteur (à condition qu'il soit aussi complet que celui d'origine), mais c'est une tâche fastidieuse, longue et propice aux erreurs.

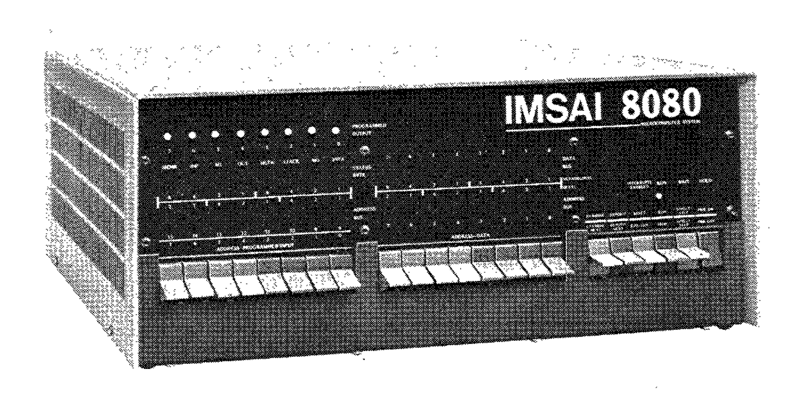



Fig. 10.11 : IMSAI traditionnel

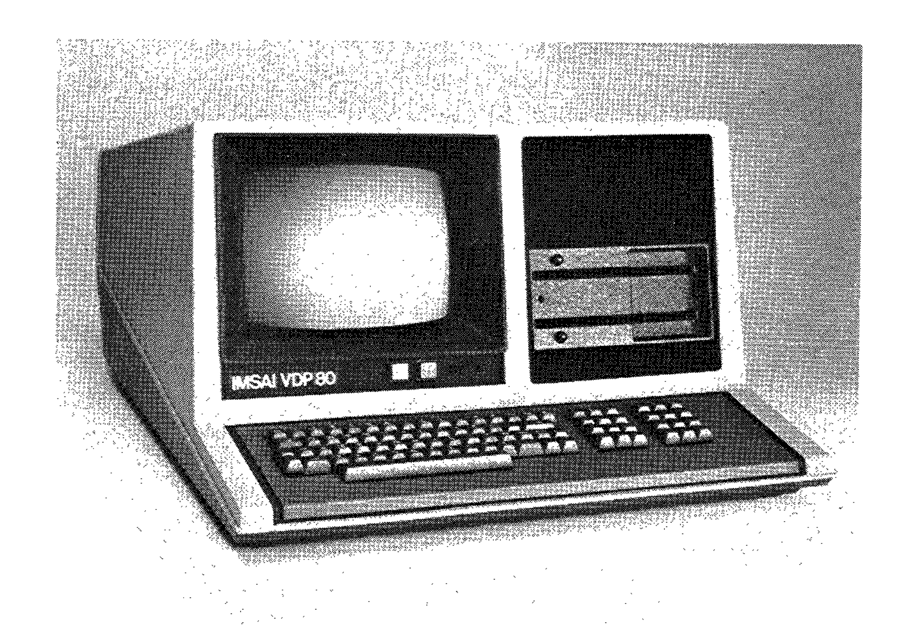



Fig. 10.12 : Version « gestion » de l'IMSAI.

### Le Sol 20 de Processor Technology

*Caractéristiques :*

■ *Microprocesseur utilisé : 8080*

■ *Mémoire : 1 K de PROM (extensible à 2 K), 2 K de RAM extensible à 64 K.*

■ *Clavier : standard style machine à écrire plus clavier numérique intégré (85 touches). Lettres majuscules et minuscules.*

■ *Entrées-sorties : sortie vidéo (exige un moniteur ou un téléviseur modifié (cf. le chapitre sur les périphériques pour voir la différence avec un téléviseur ordinaire). Comprend l'interface magnétophone (1200 bps). Connecteurs parallèle et série (RS232 et boucle 20 mA).*

■ *Particularités : bus S100. Possibilité d'ajouter cinq modules d'extension.*

Avec son clavier intégré, ce microordinateur a l'aspect des systèmes domestiques de coût minimal. Mais il a un clavier commercial complet; et assez de place pour 64 K de mémoire plus

de nombreuses cartes supplémentaires. Il est donc fonctionnellement équivalent à l'IMSAI ou à l'ALTAIR. En outre, il comprend sur une seule carte toutes les interfaces désirables pour un tel système (sauf pour le disque) sans qu'il soit nécessaire d'insérer des cartes supplémentaires.

Côté inconvénients, on ne peut ajouter que cinq cartes, ce qui peut être une limitation pour l'utilisateur qui envisage une multitude d'options. Mais il est probable que la grande majorité des utilisateurs n'ont jamais besoin de plus.

Naturellement un interpréteur BASIC 8 K est disponible, ainsi que les autres programmes usuels.

Grâce au bus S100, la plupart des modules disponibles pour l'IMSAI et l'ALTAIR peuvent être utilisés avec le SOL20.

Venu après l'Altair et l'Imsai, cet ordinateur s'est complètement débarrassé du panneau de contrôle et la conception générale est très compacte. (Il y a des interrupteurs à l'intérieur pour les nostalgiques du panneau de contrôle).

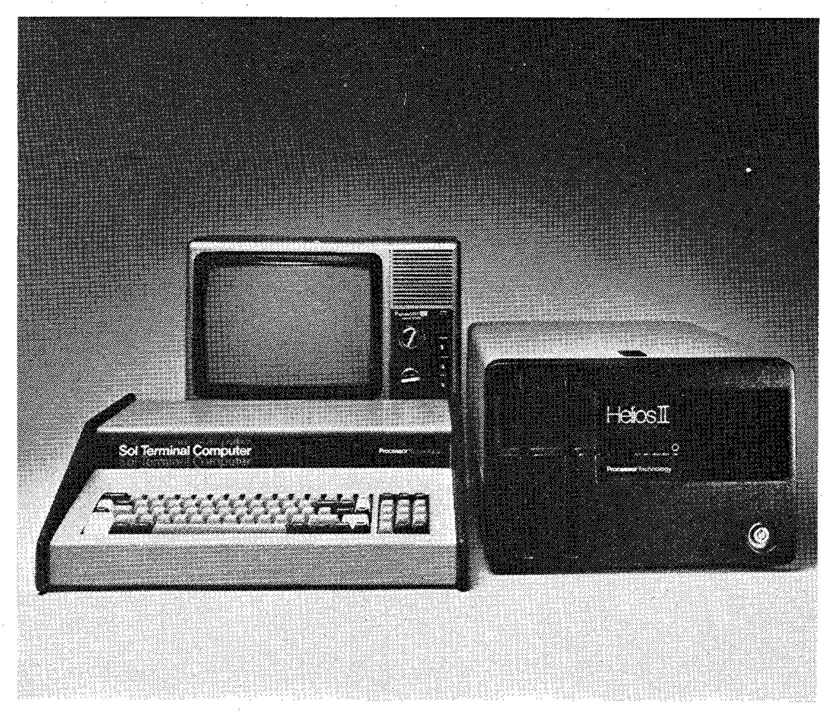



Fig. 10.13 : Le SOL a un clavier intégré type terminal.

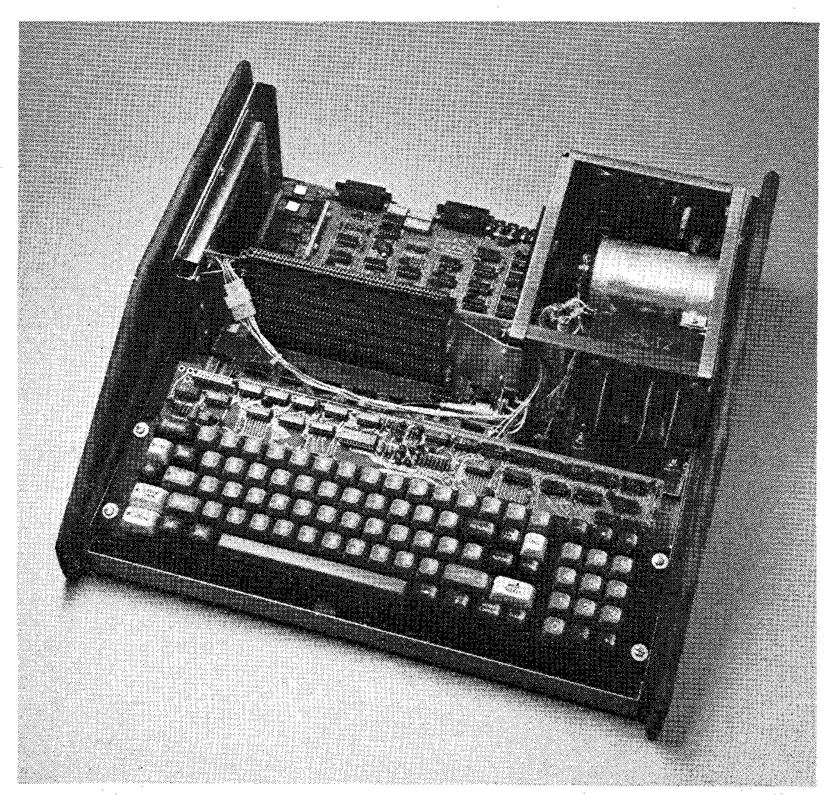


Fig. 10.14 : Le couvercle enlevé, on peut voir l'électronique du clavier (en avant), l'alimentation (à droite), et la carte microordinateur S100.

## CROMEMCO Z2

*Caractéristiques :*

■ *Microprocesseur utilisé : Z-80 (4 MHz)*

■ *Mémoire 1 K de PROM, 32 K de RAM (2 cartes 16 K), extensible.*

■ *Entrées-sorties : dépendent des cartes interfaces choisies. RS232 ou boucle de courant plus interface 8 bits en parallèle.*

■ *Particularités : utilise le bus S100 ; 21 emplacements de cartes. Disponible en coffret intégré avec une ou deux unités de disques souples.*

Ce microordinateur appartient à la famille des produits compatibles bus S100. Le microprocesseur qu'il utilise est plus rapide que le 8080 standard mais il n'en résulte pas forcément (cf le chapitre sur le choix d'un système) que le système soit globalement plus performant. Les temps d'exécution du software doivent être contrôlés sur un système muni de disques, pour chaque programme. Le système a — potentiellement — des performances améliorées, si le programme est bien conçu. Le software comprend un BASIC étendu 16 K et un compilateur FORTRAN IV. Il y a aussi, disponible sur option, un panneau de contrôle (presqu'identique à celui de l'IMSAI 8080). Une interface télévision couleur en deux cartes, appelée « dazzler » ( = « qui éblouit ») est disponible ainsi qu'une console manche à balai avec haut-parleur.

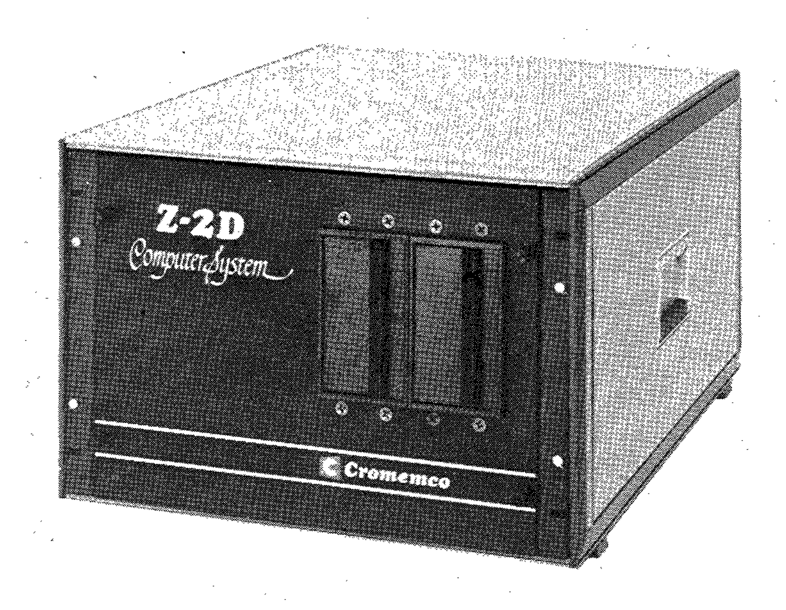



Fig. 10.15 : Z2 avec l'option double disques.

Fig. 10.16 : Le « Dazzler » fait l'interface avec un téléviseur couleur ; il comporte deux cartes.

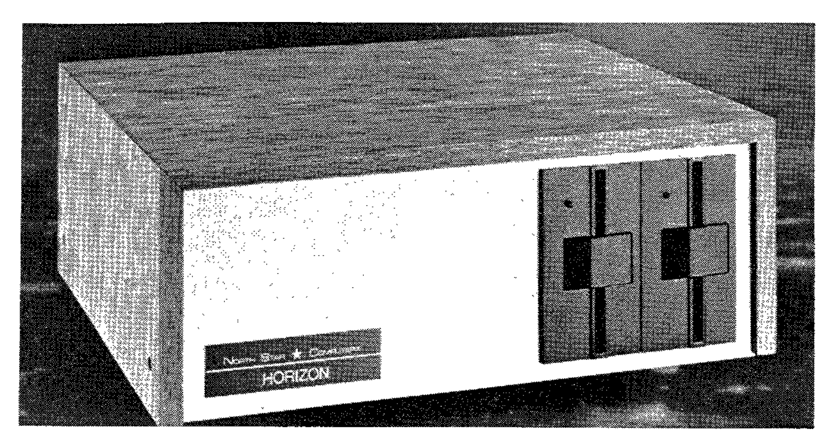


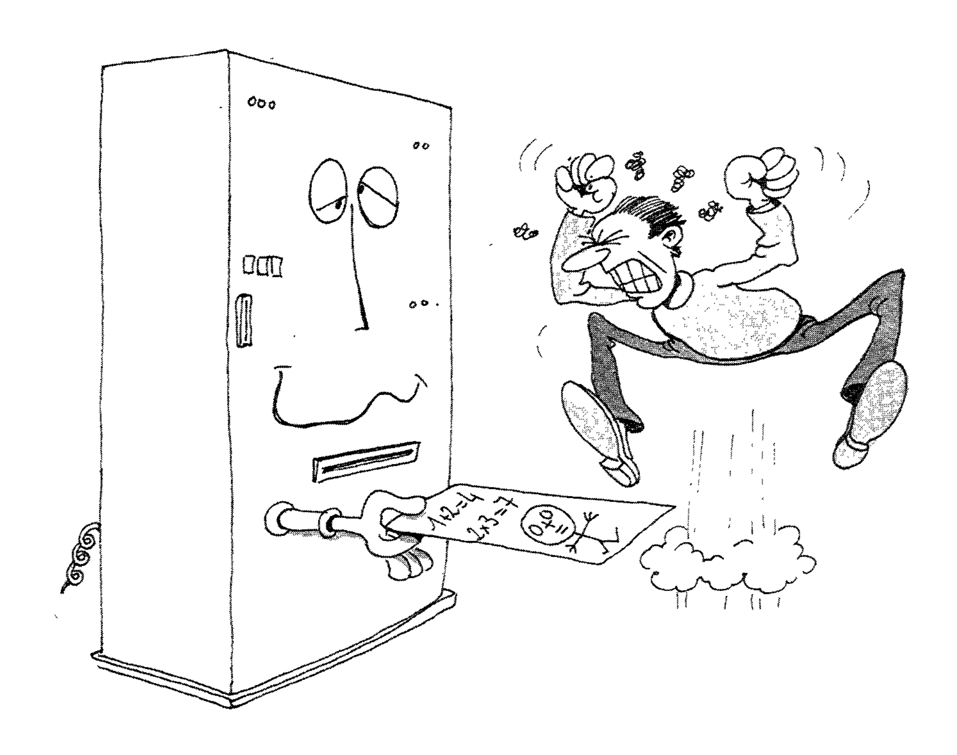

Fig. 10.17 : L'Horizon de Worth Star.

## L'Horizon de North Star

*Caractéristiques :*

- *Microprocesseur utilisé : Z-80 (4 MHz)*
- *Mémoire : extensible jusqu'à 64 K*
- *Particularités : double unité de disques intégrées. Bus S100.*

Ce système est, lui aussi, compatible S100. Il a un bon BASIC complet et il fait appel à la version rapide du Z-80. Une carte arithmétique flottante est disponible pour les calculs rapides.




Enseignement scientifique.

**L'Equinox de Parasitic Engineering**

*Caractéristiques :*

- *Microprocesseur utilisé : 8080*
- *Particularités : bus S100, carte spéciale panneau de contrôle. Le panneau de contrôle a un clavier octal et un affichage à LEDS. Il permet le pas-à-pas.*

**Le Xitan de Technical Design Labs**

*Caractéristiques :*

- *Microprocesseur utilisé : Z-80*
- *Mémoire extensible jusqu'à 64 K*
- *Entrées-sorties : deux portes série, interface cassette 1200 baud.*
- *Particularités : bus S100.*

**Le Challenger de Ohio Scientific**

Ohio Scientific offre une gamme de systèmes utilisant le 6502 (1), le Z-80 et le 6800 sur une même carte, ainsi que des cartes d'interface variées. En outre, un logiciel étendu est disponible.

**Le Vector 1 de Vector Graphic**

*Caractéristiques :*

- *Microprocesseur utilisé : 8080*
- *Mémoire : extensible jusqu'à 64 K*
- *Particularités : bus S100.*

---

(1) Il s'agit d'un 6502 rapide à 2 MHz

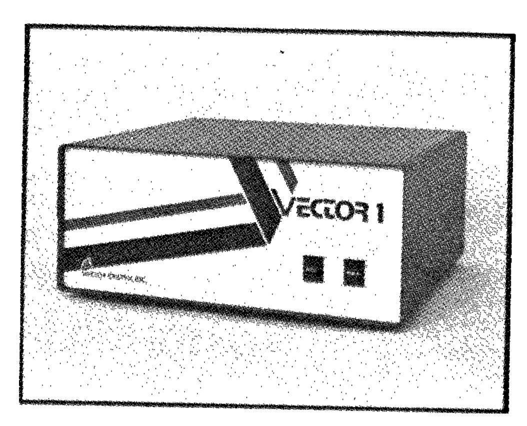



Fig. 10.18 : Le Vector 1.

**Polymorphic 88**

*Caractéristiques :*

■ *Microprocesseur utilisé : Z-80*

■ *Mémoire : 3 K de ROM, 16 K de RAM, extensible jusqu'à 64 K.*

■ *Particularités : Bus S100 (5 emplacements). Plusieurs options de présentation disponibles.*

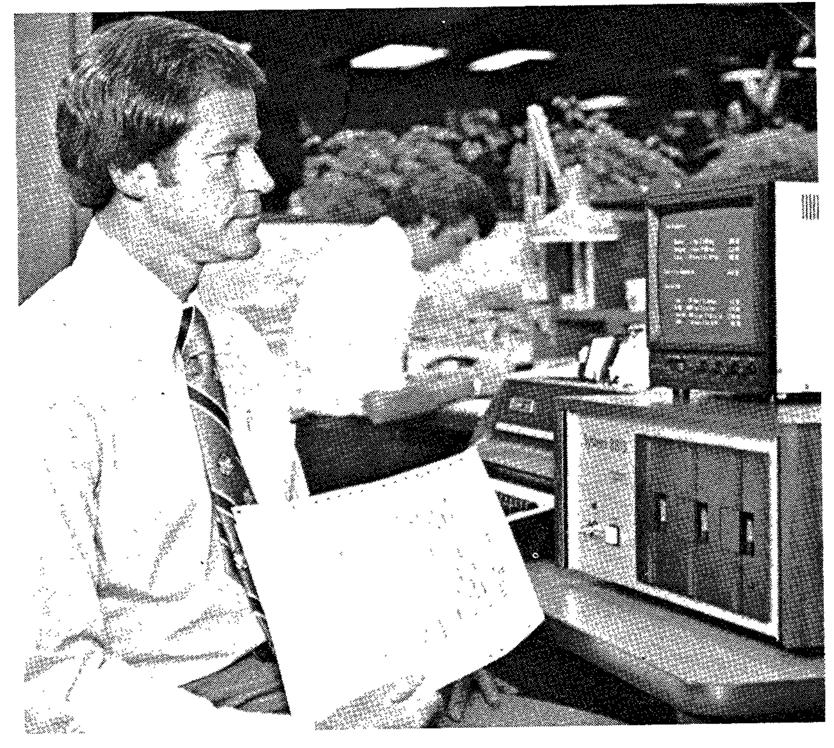


Fig. 10.19 : Polymorphic 88.

## Heathkit : H8

*Caractéristiques*

- *Microprocesseur utilisé : 8080*
- *Mémoire : 4 K de RAM extensible à 16 K sur la carte*
- *Particularités : clavier hexadécimal et LEDS.*

Le H8 est le bas de gamme de Heathkit sur le marché des microordinateurs. Des périphériques variés se connectent facilement au système.




Fig. 10.20 : Heathkit H8.

### Heathkit H11

*Caractéristiques :*

- *Microprocesseur utilisé : LSI 11*
- *Mémoire : totalement extensible jusqu'à 64 K*
- *Particularités : utilise le bus standard LSI 11-Q. Peut utiliser le software du PDP 11 grâce à un accord de licence avec Digital Equipment.*

Une des graves lacunes des microordinateurs était le manque d'une vaste bibliothèque de programmes utilisables. Le PDP 11 est l'un des miniordinateurs pour lesquels il existe la plus vaste bibliothèque de programmes. Le H11 représente une tentative d'en profiter. Il fait appel au jeu de boîtiers LSI 11 qui émule un PDP 11/03, c'est-à-dire qu'il accepte le même jeu d'instructions : on peut exécuter sur le H11 les programmes du PDP 11/03. En outre on peut facilement lui connecter les périphériques standards du LSI 11. Précaution : le bus « LSI11-Q » n'est pas identique à l'« Unibus » du PDP 11.

Ce système devrait être séduisant pour tous ceux qui ont utilisé la bibliothèque de software du PDP 11, ou qui y ont accès.

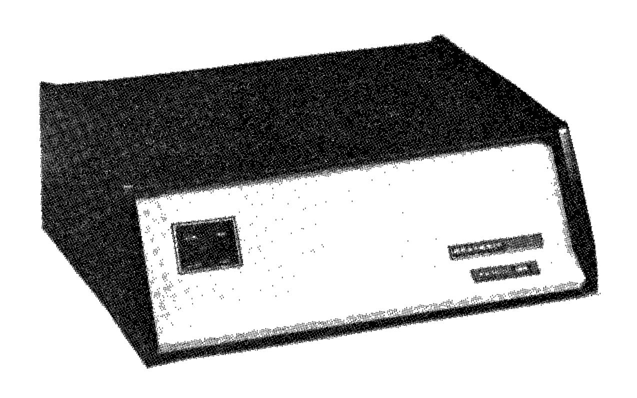


Fig. 10.21 : Heathkit H11.

### Le Sord

*Caractéristiques :*

- *Microprocesseur utilisé : Z-80*
- *Mémoire : extensible jusqu'à 64 K et plus*

■ *Entrées-sorties : visu intégrée 30 cm (1920 caractères), imprimante 40 colonnes intégrées, unité de minidisquette et clavier étendu.*

■ *Avantages : système totalement intégré ; bus S100.*

Ce système complètement intégré est un des premiers qui soit présenté par les Japonais dans ce domaine.

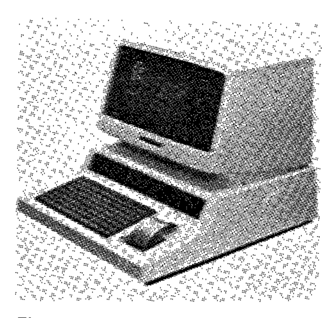


Fig. 10.22 : Le Sord (Japonais).

**Le Bytemaster de Digital Group**

*Caractéristiques :*

■ *Microprocesseur utilisé : Z-80*

■ *Mémoire : de 18 à 64 K selon l'unité centrale*

■ *Entrées-sorties : clavier complet ; visu intégrée (9 pouces, 16 lignes de 64 caractères, majuscules et minuscules). Unité de minidisquette double face (ou cassette digitale).*

C'est un système de faible encombrement, complètement intégré, avec un clavier détachable sous forme de machine de table. Il est disponible aussi sous forme de modules séparés. L'unité centrale accepte 64 K de mémoire et des périphériques externes supplémentaires. Tous les périphériques ont une carte d'interface qui doit être insérée à l'emplacement approprié du fond de panier de l'unité centrale. Ce n'est pas un système S100.



Fig. 10.23 : Système de Digital Group.

**Compucolor II**

*Caractéristiques :*

■ *Microprocesseur utilisé : 8080A*

■ *Mémoire : 16 K de ROM, 8 K de RAM (y compris 4 K de rafraîchissement d'écran), extensible à 32 K de ROM, 32 K de RAM.*

■ *Entrées-sorties : visualisation couleur intégrée (13 pouces, 32 lignes de 64 caractères) ; clavier standard ASCII sur option ; minidisquette intégrée, port RS232C.*

■ *Particularités : software graphique avec tracé de vecteurs, 64 caractères spéciaux.*

Ce système offre à la fois du traitement graphique couleur limité et des possibilités de calculs généraux.

**Digital Equipment PDP 11/03**

*Caractéristiques :*

■ *Microprocesseur utilisé : jeu de boîtiers LSI 11.*

■ *Mémoire : 4 K extensible à 64 K*

■ *Avantages : compatible avec le jeu d'instructions de la ligne PDP11. Excellent software. Maximum de 6 modules additionnels dans le coffret.*

Ce microordinateur a été développé par DEC comme remplacement bon marché du miniordinateur 11/05 et il utilisait au départ le jeu de boîtiers LSI 11 de Western Digital. Ce système peut être utilisé comme ordinateur personnel mais il a surtout été mis sur le

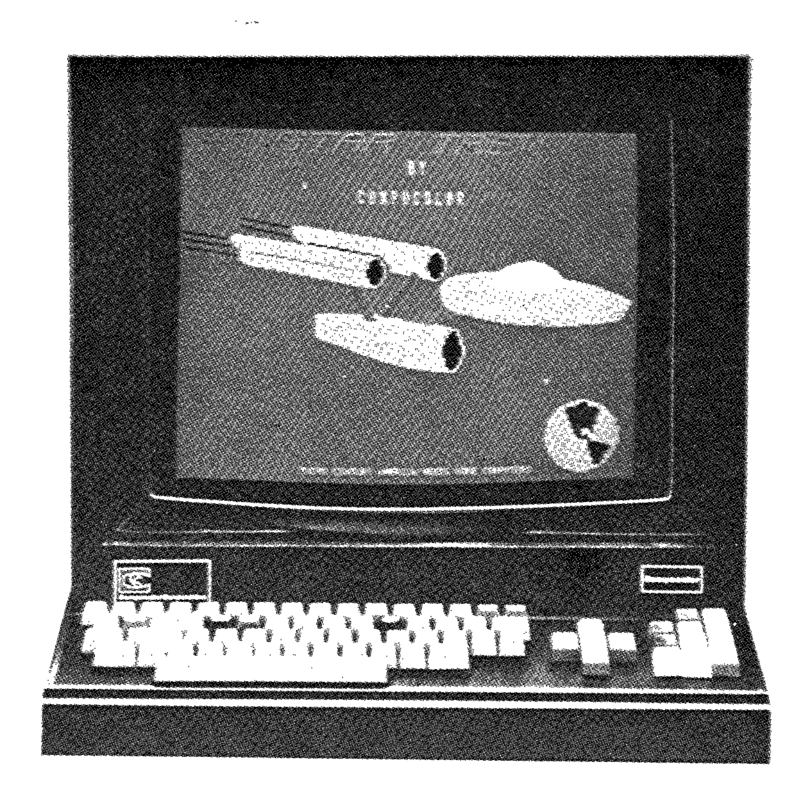



Fig. 10.24 : Le Compucolor a un écran couleur intégré. Le modèle montré ici a disque séparé.

marché dans les branches traditionnelles de DEC, c'est-à-dire les laboratoires et le contrôle des processus.

Comme tous les PDP 11, il a un software étendu, en particulier un BASIC niveau Dartmouth, FORTRAN ANSI et un système d'exploitation temps réel avec programmes d'avant-plan et arrière-plan simultanés : RT11.

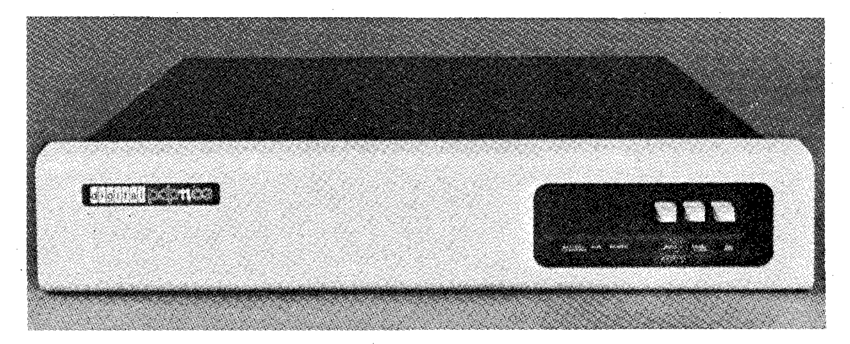



Fig. 10.25 : Digital PDP 11/03.

## Le SWTPC 68/2 de South West Technical Products

Système intégré avec clavier style machine à écrire et écran. Il utilise le microprocesseur 6800 et a défini le bus SS-50.

## Alpha Micro Systems

Ce microordinateur permet les traitements sur 16 bits puisqu'il utilise le microprocesseur WD-16 de Western Digital. Il est compatible S100. Il est équipé d'un remarquable système d'exploitation (temps partagé et gestion de fichiers).

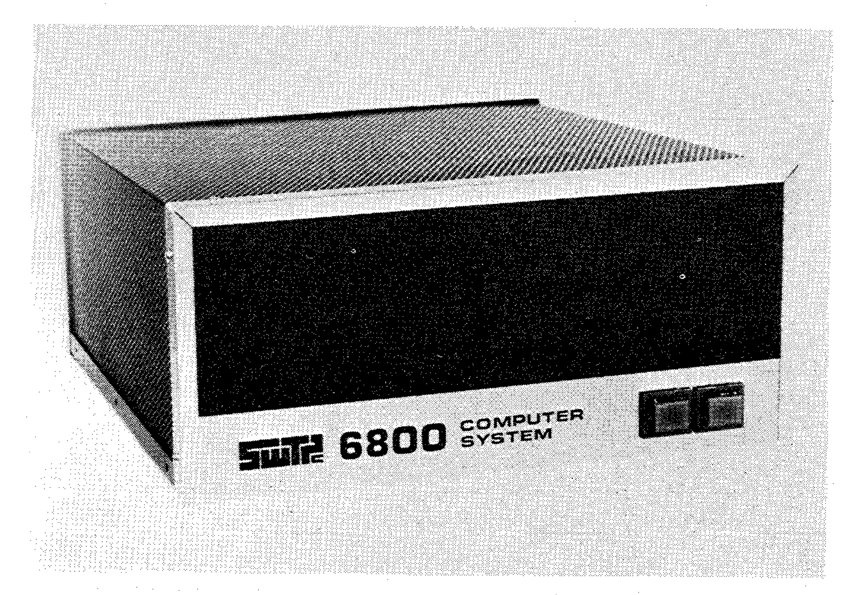



Fig. 10.26: SWTP 6800.

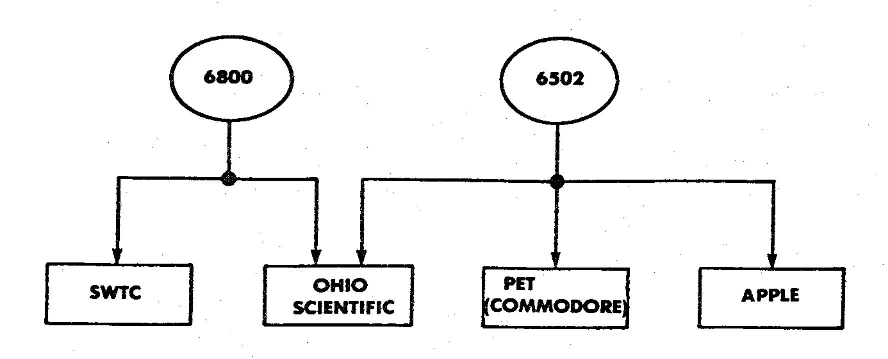



Fig. 10.27: Microordinateurs des familles 6800 et 6502.

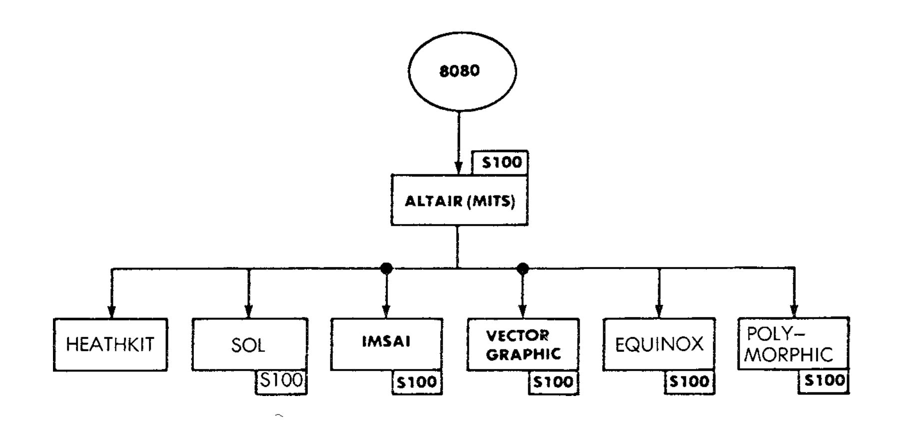



Fig. 10.28: Microordinateurs de la famille 8080.

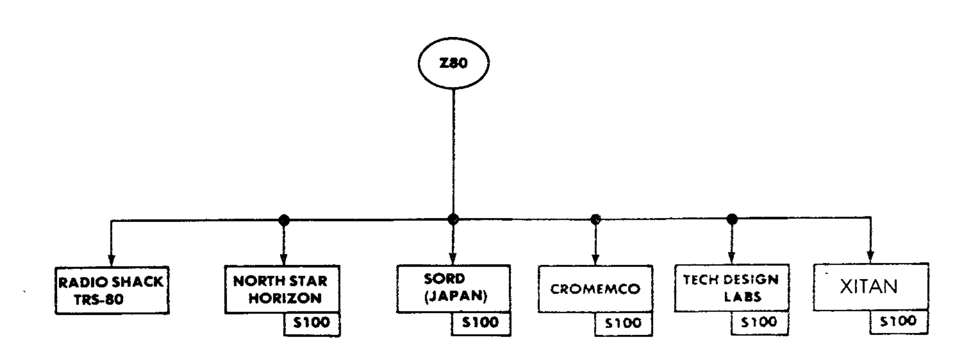



Fig. 10.29: Microordinateurs de la famille Z-80.

## PETITS SYSTEMES DE GESTION

Un certain nombre de constructeurs font maintenant des systèmes de gestion qui utilisent un microprocesseur, ou non. Une liste des constructeurs traditionnels de petits systèmes de gestion apparaît en fin d'ouvrage (appendice E).

IBM a introduit en 1978 le 5110, de conception propre, de prix voisin des microordinateurs qui fonctionne en BASIC et en APL et a imprimante, bande et disquette intégrées. Il a un clavier machine à écrire normal.

## CP/M

Il est important de souligner que, dès qu'un système est équipé d'un disque, il nécessite un logiciel de gestion de disque, le « DOS » (Disk Operating System). Le DOS va, en particulier, formater les fichiers disques.

A moins que le DOS ne soit le même, les programmes pour deux microordinateurs différents sont alors incompatibles s'ils utilisent le disque.

Répétons : bien que les deux microordinateurs utilisent par exemple le 8080, un programme créé pour le premier ne fonctionnera pas sur le second, en raison d'une incompatibilité de système de fichiers.

Dans le cas des microordinateurs compatibles S100, un système d'exploitation DOS est maintenant devenu standard : c'est CP/M (marque brevetée). CP/M est maintenant disponible pour presque tous les microordinateurs de type S100 et pour presque toutes les unités de disque.

Les programmes écrits sous CP/M sont alors complètement compatibles. CP/M requiert un (ou plusieurs) disques 8 pouces (en principe pas les minidisquettes). La plupart des programmes de gestion sont donc disponibles maintenant en version CP/M. Ils marcheront sur n'importe quel système exécutant CP/M.

## RECAPITULATION

Les microordinateurs ont été classés en trois catégories : carte (formation), système individuel intégré (distraction et traitements limités) et systèmes d'usage général. Dans cette dernière catégorie, il

existe aujourd'hui un grand nombre de produits qui offrent différentes combinaisons à l'intérieur du coffret. Les critères essentiels sont :

**Pour une utilisation de gestion :** seuls conviennent les microordinateurs possédant :

— mémoire complète (jusqu'à 64 K au total) dans le coffret.

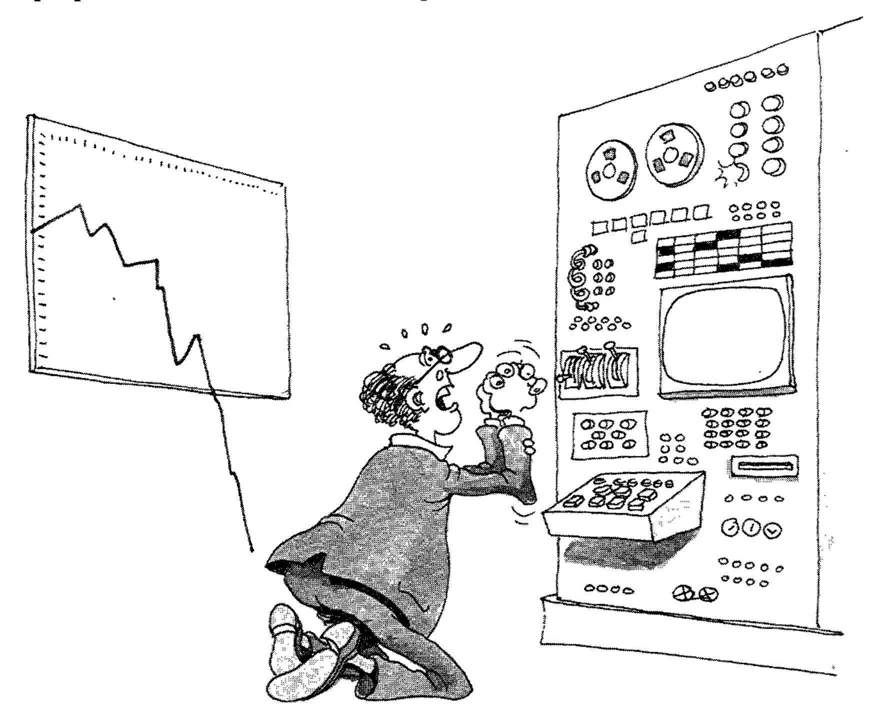
— BASIC complet de préférence pouvant résider en ROM (pour raisons d'efficacité).

— ensemble de périphériques complet, ou alors bus standard auquel on peut connecter des périphériques extérieurs.

— logiciel complet comprenant un DOS (système d'exploitation disque) et un système de gestion de fichiers.

— programmes de gestion disponibles capables de tourner directement sur le processeur considéré.

**Pour l'utilisation individuelle :** la plupart des systèmes ont des performances analogues ; ils diffèrent sur le plan des options d'entrées-sorties. Les données comparatives fournies par l'analyse qui précède devraient être utiles pour finaliser le choix.



# CHAPITRE XI
# COUT D'UN SYSTEME DE GESTION

## LE PRIX VERITABLE D'UN SYSTEME

Il est possible aujourd'hui de s'amuser avec un système minimal pour 1 500 à 4 000 F, alimentation comprise et avec des entrées-sorties minimales.

Mais si l'on a l'intention de faire de la « programmation sérieuse » ou de faire tourner des programmes de gestion, il faudra généralement :

1. Ajouter de la mémoire.
2. Ajouter un terminal à écran.
3. Ajouter un ou deux disques (s'ils ne sont pas compris).
4. Ajouter une imprimante (presque jamais comprise).

Partant d'un coffret unité centrale de 7 à 15 000 francs, on arrivera à un prix du système global de 4 fois le prix de l'UC ou même plus dans le cas d'un système de gestion (il s'ajoute, en plus, le coût du logiciel).

Ceci a toujours été vrai de tous les ordinateurs : dans un système complet, le coût dominant est celui des périphériques.

Bien que le coût des composants électroniques doive continuer à décroître, il est peu probable que le prix des périphériques diminue considérablement en raison de leur nature électromécanique, sauf s'il apparaît un nouveau marché de masse.

### Les coûts invisibles

Au prix d'achat d'un système s'ajoutent deux facteurs principaux : les frais de maintenance et le coût de la programmation.

Pour un système de gestion, la règle générale est d'estimer les frais mensuels de maintenance à 1 % du prix d'achat.

La programmation ne conduit pas à des frais significatifs dans le cas d'un système personnel : on programme par plaisir, ou bien on achète des programmes pour pas cher.

Dans le cas d'un système de gestion, les programmes standards de gestion sont précisément cela : ils sont standardisés et ne sont donc jamais exactement adaptés à tous les besoins, particulièrement à ceux du petit commerce qui tendent à être très individualisés.

Il est habituel de prévoir au moins un programmeur à mi-temps dans le coût permanent d'un système d'informatique. Le programmeur aura la responsabilité d'adapter les programmes standards et de développer les programmes spécialisés requis par l'application.

Un troisième coût invisible qui s'ajoute aux précédents résulte de l'impact des procédures informatisées sur la gestion. Dans un cas idéal, l'ordinateur élimine la nécessité de certains personnels et simplifie considérablement un grand nombre de tâches.

En pratique, ce n'est pas toujours le cas. Dans une petite entreprise, il est peu probable que l'ordinateur supprime des emplois. On aura toujours besoin de la personne qui tient les livres comptables et des vendeurs. Toutefois, l'ordinateur automatise leurs tâches, ce qui gagne du temps, mais, surtout il rend d'autres services :

— mises à jour immédiates,
— gestion d'inventaire,
— statistiques et rapports immédiats.

Les personnels qualifiés auront la plupart de leur temps libre pour des tâches plus complexes ; quant aux tâches répétitives (statistiques de vente, notices de rappel, recommandes automatiques) elles seront effectuées avec plus d'exactitude.

En situation de croissance, le microordinateur peut probablement éliminer la nécessité de personnel de bureau supplémentaire, ce qui économise des salaires.

Enfin, la disponibilité d'informations parfaitement à jour est de grande valeur pour la direction.

En bref, dans tous les cas où les critères précédents sont satisfaits, c'est-à-dire croissance ou valeur importante d'une information instantanée, un ordinateur entraînera des économies immédiates.

Dans d'autres cas, son utilisation peut ne pas être nécessaire, sauf si d'autres services présentent un intérêt particulier comme, par exemple, la gestion de fichiers d'adresses.

**Quand acheter ?**

Soit P le prix d'achat du système,
M le coût de la maintenance,
A les frais supplémentaires (programmeur ou autres),
Le coût du système est $P + (M + A) \times T$ où T est le temps (en mois).

En fait, il faut ajouter $L$ = coût d'installation du système et de passage à l'informatique (il n'intervient qu'une fois).

Le coût total est alors : $P + (M + A) \times T + L$

Du côté des économies, l'ordinateur est supposé économiser D francs par mois par réduction du personnel et suppression des traitements informatiques extérieurs (liste d'adresses, paie...).

Si D est faible, le système ne sera pas « rentable de lui-même » sauf si l'on prend en considération la valeur chiffrée des services supplémentaires qu'il rend.

Si D est important, le système se sera amorti de lui-même quand son prix aura été compensé par les économies c'est-à-dire quand :

(1) $P + (M + A) \times T + L = D \times T$

c'est-à-dire après le temps :

$$T = \frac{P + L}{D\text{-}M\text{-}A}$$

Si l'on veut que ceci se produise au bout de 18 mois, il faut que les économies mensuelles soient au moins :

$$D = \frac{P + L + (M + A) \times 18}{18}$$

Enfin, il faudrait inclure dans D, non seulement les réductions de salaires et les économies sur les traitements informatiques extérieurs mais aussi le revenu supplémentaire que représentent :

— la disponibilité de rapports à jour,

— les réapprovisionnements immédiats (pas de pertes dues à des délais),

— un meilleur marketing grâce à l'analyse des ventes et la gestion de la liste d'adresses,

— une meilleure collecte des comptes à recevoir.

La formule proposée en (1) peut donc probablement être considérée comme une hypothèse basse.

En outre, la valeur éducative du processus d'informatisation entraînera des économies importantes quelques années plus tard, lorsque la société se sera développée.

### Y-a-t-il des alternatives ?

Les alternatives qui s'offrent traditionnellement à l'introduction d'un système dans l'entreprise sont :

1. traitements informatiques à façon à l'extérieur,
2. location d'un terminal de temps partagé.

Examinons-les.

### Traitement informatique à l'extérieur

Pour des tâches très spécialisées, le traitement à l'extérieur peut s'avérer économique : paie, impôts, liste d'adresses. Cependant, il présente des inconvénients : coût qui s'accroît si l'entreprise se développe, parfois problème de fiabilité, de temps de réponse, de sécurité de l'information, de disponibilité de l'information.

Le traitement à l'extérieur reste une possibilité viable pour de nombreuses entreprises qui ne tireraient qu'un avantage limité à avoir un ordinateur sur place.

De fait, il apparaît des sociétés de service qui font appel à un microordinateur pour fournir des services spécialisés à un groupe d'utilisateurs tels que des P.M.E.

### Temps partagé

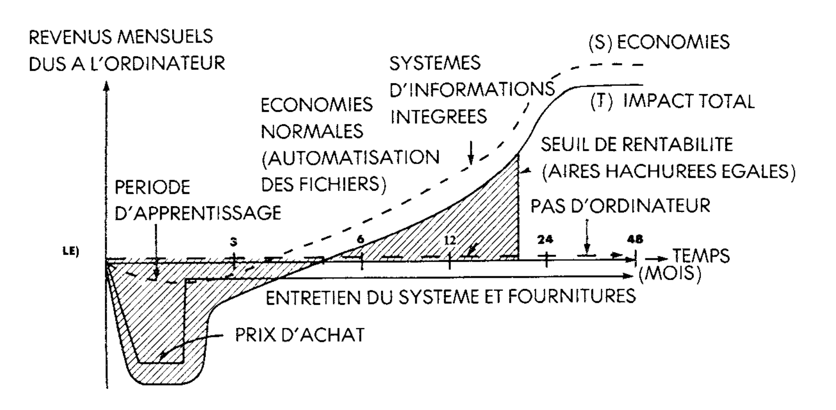
Il est possible de louer un terminal (à écran ou à imprimante) et de communiquer avec un ordinateur central par l'entremise d'une ligne téléphonique. L'utilisateur paie la location du terminal, la ligne, le temps de connexion et le temps de traitement utilisé. Cette option est très commode : on est relié à un ordinateur puissant, muni de toutes les facilités. Mais c'est une option chère : le prix de location ou d'achat d'un terminal n'est pas négligeable devant le prix d'un microordinateur.

Sauf si le traitement dont on a besoin est très spécialisé (ou bien si l'accès à l'ordinateur central est gratuit), un système sur place est maintenant plus économique (à condition de pouvoir rendre les services dont on a besoin).

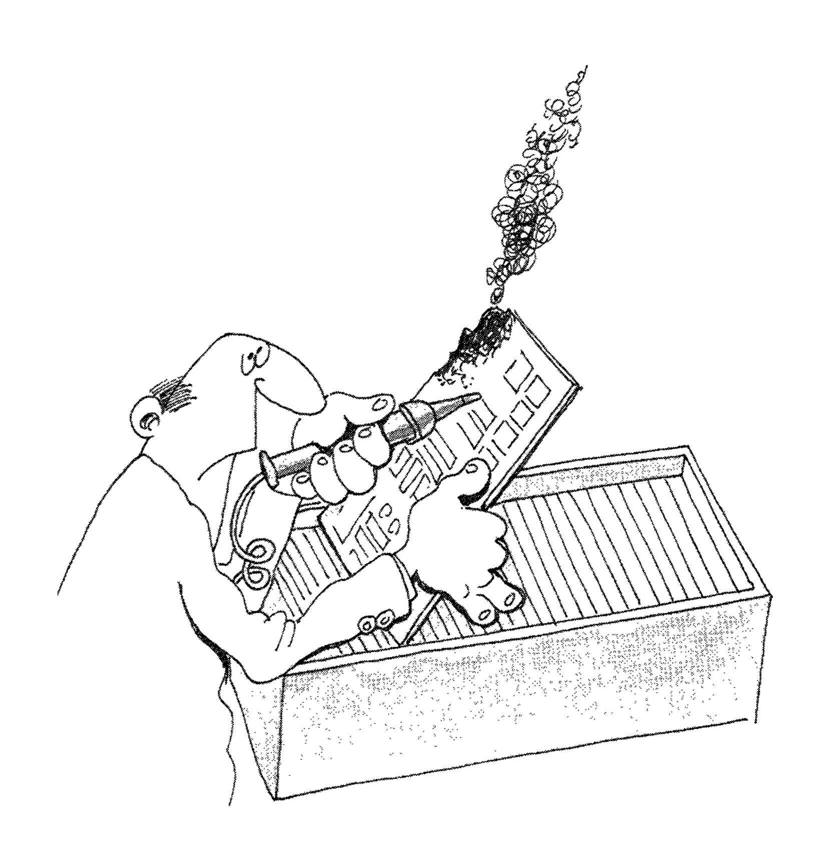
## RECAPITULATION

On vient d'analyser et d'évaluer les équilibres coûts-bénéfices dans un système microordinateurs. Dans la plupart des entreprises en croissance il est probable que le seuil de rentabilité de l'ordinateur soit vite atteint. Si l'on dispose du capital nécessaire, le fait de s'équiper au plus vite représente un investissement rentable et un avantage ultérieur important.

Enfin, sous réserve de quelques précautions, les risques financiers sont limités.



# CHAPITRE XII
# COMMENT ECHOUER AVEC UN SYSTEME DE GESTION



## INTRODUCTION

Les échecs vont être considérés comme accidentels, c'est-à-dire dus à des erreurs. Ce chapitre va explorer les sources d'erreurs possibles. Tout utilisateur d'un système de gestion doit être au courant des pièges pour pouvoir les éviter.

Le fil conducteur fondamental des commentaires qui suivent est qu'un système de gestion doit être *sûr et fiable*. Les erreurs sont imputables à trois sources principales : le matériel, le logiciel et les procédures.

## PANNES HARDWARE

On suppose que le système a été livré en bon état de marche. Les 100 à 200 premières heures de fonctionnement s'appellent la période de « burn in » ou rôdage. C'est pendant cette période que les composants défectueux ont la plus grande probabilité de lâcher. De nombreux constructeurs ou revendeurs effectuent le rôdage des systèmes avant de les expédier.

Considérons maintenant les pannes les plus fréquentes au cours de la vie de l'équipement.

### Pannes mécaniques

Les pannes mécaniques sont les plus fréquentes et la plus grande candidate à la panne est l'imprimante. Pour éviter de telles pannes, on a, en production, recours à la maintenance préventive : les pièces mécaniques sont examinées, nettoyées et ajustées toutes les X heures. Cette procédure empêche la plupart des malfonctions avant qu'elles ne se produisent.

### Pannes dûes à l'environnement

Tous les composants d'un système sont prévus pour une gamme de température et un niveau d'humidité déterminés. Il est évident que les spécifications doivent être satisfaites. Toutefois, les malfonctions dues à un non respect temporaire des conditions sont généralement transitoires et elles n'entraînent pas de détérioration permanente du système.

### Pannes électroniques

Comme dans tout système complexe, certains composants électroniques peuvent mal fonctionner. On recommande généralement de laisser le diagnostic et la réparation aux mains du fournisseur. Ceci implique la présence d'un service de réparations local efficace. Pour des détails sur les techniques de détection des pannes, le lecteur technicien peut se reporter à notre référence C5.

### Récapitulation des pannes hardware

Si le système est construit en conformité aux normes généralement acceptées pour un fonctionnement fiable, ou, en pratique, s'il est réputé fiable, il est probable que les pannes hardware seront négligeables après les tout premiers jours. Elles sont normalement confiées au service après-vente du constructeur ou bien au fournisseur local. Il est d'importance cruciale pour un système de gestion qu'il y ait un service de réparation local.

### Techniques particulières pour l'amélioration de la fiabilité

La *parité* a été longtemps une technique favorite pour vérifier la transmission ou la conservation correcte de l'information. La parité consiste à ajouter un bit supplémentaire à chaque octet de données afin d'en contrôler le contenu. En parité *paire,* on ajoute un « 0 » si le nombre total de « 1 » dans l'octet est pair, et un « 1 » dans le cas contraire. Autrement dit, on assure que le nombre total de « 1 » sera pair. (On peut aussi fonctionner en parité *impaire*).

Cette technique décèle les erreurs portant sur un bit, qui sont les plus fréquentes : si un seul bit a changé d'état, le module de contrôle de parité compare le bit de parité calculé avec celui qui est rangé avec l'octet considéré et il détecte l'erreur.

La parité est très utilisée dans les systèmes moyens et importants. Elle n'est presque jamais utilisée dans les microprocesseurs.

Il y a à cela deux raisons :

1. Les systèmes à microprocesseurs sont bien plus fiables que les ordinateurs traditionnels, même minis, simplement parce qu'ils ont beaucoup moins de composants (le taux de panne augmente très vite avec le nombre de composants et de liaisons).

2. Il n'y a pas eu jusqu'à présent de demande pour une augmentation de la fiabilité au prix d'un accroissement du coût et de la complexité.

Un des domaines où les erreurs risquent de se produire est la mémoire du système, principalement parce qu'elle contient beaucoup de composants. Une erreur transitoire durant un jeu de guerre de l'espace n'est pas grave : on redémarre le jeu. Une erreur temporaire qui efface des données de gestion qu'il a fallu des heures pour collecter est, elle, rédhibitoire.

Comme, en outre, ce sont les systèmes de gestion qui ont le plus de chances de nécessiter les plus larges mémoires RAM disponibles (40 ou 48K octets), ce sont eux qui risquent le plus les erreurs mémoire. Il va probablement bientôt y avoir des systèmes de mémoire pour microprocesseurs munis, sur option, de la parité pour procurer un supplément de fiabilité.

La parité est définie au niveau de l'octet. Cependant, dans le cas de mémoires de masse telles que disques ou bandes, il n'est pas possible de consacrer à cette fonction un bit supplémentaire tous les 8 bits. Dans ce cas, on utilise un octet entier (ou plusieurs) à la fin de chaque bloc de longueur déterminée. Cet octet contient une « somme de contrôle » ou un CRC (contrôle de redondance cyclique).

*La somme de contrôle* est calculée à partir d'une formule simple qui fait intervenir les n octets précédents (n dépend du système). Si un octet a changé, la somme de contrôle change et le module de vérification décèlera l'erreur à la lecture du bloc de données. Le CRC fait appel à une technique plus complexe pour calculer l'octet de contrôle.

Les sommes de contrôle et le CRC sont universellement utilisés dans le cas des disques ou des bandes. La somme de contrôle est une méthode simplifiée, le CRC est une méthode plus fiable.

Enfin, une bonne pratique est de *relire après avoir écrit :* à chaque fois qu'on a écrit un bloc de données, il faudrait le relire. C'est quelquefois fait dans le cas de disques et devrait l'être pour les fichiers cruciaux. C'est une fonction software. Habituellement, elle n'est pas réalisée car elle ralentit les opérations d'écriture. Il serait intéressant qu'elle soit disponible en option.

## ERREURS SOFTWARE

*Tout programme complexe devrait être supposé erroné !* Au moins

au sens mathématique : il y aura toujours une combinaison d'événements qui fera tomber le programme en panne. Néanmoins, la probabilité est faible et on peut espérer qu'il n'y aura pas de dommages.

Comme il n'y a aucun moyen de garantir qu'un long programme — comme toute construction humaine complexe — est totalement dépourvu d'erreurs, tout système aura occasionnellement des pannes software.

Traditionnellement, les constructeurs diffusent des mises à jour périodiques de leurs programmes, où les erreurs qui ont été décelées sont corrigées.

Par exemple, lors de sa première diffusion presque tout interpréteur BASIC se comporte incorrectement vis-à-vis de certaines séquences d'instructions. Ces problèmes sont corrigés ensuite.

Tout software qui n'a pas encore été testé par un grand nombre d'utilisateurs doit être supposé contenir des erreurs. Tout nouveau software doit donc être considéré avec précautions dans un environnement de gestion.

Cependant, il ne faut pas non plus trop s'inquiéter : certains utilisateurs ne constateront jamais de malfonctions décelables. Si le système est bien conçu, les malfonctions sont souvent limitées à la nécessité de répéter une opération qui n'a pas marché.

Même les plus gros systèmes IBM subissent des incidents catastrophiques à l'occasion. En fait, à cause de leur complexité même, ils risquent fortement de tomber en panne et ils nécessitent un grand nombre de techniques d'amélioration de la fiabilité.

## PROCEDURES D'UTILISATION

Il faut distinguer plusieurs cas vu que c'est probablement le domaine où on peut trouver les erreurs les plus sérieuses.

Comme il est difficile de discriminer strictement certaines procédures de leur réalisation software, plusieurs aspects software vont aussi être discutés ici.

### Exactitude des données

Les données doivent être vérifiées au moment où elles sont introduites et on doit maintenir leur intégrité tout au long du traitement.

*A l'entrée, il faut vérifier l'exactitude des données.* Dans beaucoup de cas, cela peut se faire par contrôles de zones ou contrôles de limites.

*Un contrôle de zone* vérifie par exemple, qu'une zone « numérique » ne contient pas par accident des lettres ou deux virgules.

*Un contrôle de limites* est une forme de « test de vraisemblance ». Par exemple, un numéro de mois doit être compris entre 1 et 12. Une donnée inhabituellement grande ou petite doit être détectée. Une horrible histoire d'ordinateurs qu'on raconte traditionnellement est la suivante : une perforeuse inexpérimentée (ou surmenée) d'un bureau municipal, tape un jour une commande de 1 000 000 de dollars dans le fichier des comptes à recevoir, au lieu de 10 000 dollars. Le responsable a accepté un taux de dépenses de 30 % des recettes. Avant que l'erreur ne soit décelée, 300 000 dollars allaient être dépensés.

Une autre histoire catastrophique est celle de M. Dupont. On reçoit un paiement de 4 000 F de M. Dupont. On crédite immédiatement son compte qui montrait un dû de seulement 1 000 F. Un chèque de remboursement de 3 000 F est émis. Un mois plus tard, M. Dupont furieux téléphone et demande pourquoi on lui envoie un « dernier avertissement » de la part de l'ordinateur alors qu'il a tout payé. Vous avez compris pourquoi. Il y avait deux M. Dupont dans le fichier et on avait crédité l'autre.

Toute procédure humaine ou automatisée devrait vérifier une telle situation. Au mieux, le programme signale la situation des « Dupont multiples ».

**Renforcement des contrôles**

Un principe général en gestion est de séparer les tâches : la personne qui vérifie les données ne doit pas être la même que celle qui les a entrées.

Cela réduit le risque d'erreurs accidentelles, et aussi introduit un contrôle contre les modifications par malveillance ou autres causes.

En outre, au point de vue des procédures, c'est une pratique de gestion vitale que de conserver un chemin de vérification dans tout fichier : simplement tous les fichiers et toutes les transactions doivent contenir les références appropriées aux fichiers d'origine ou aux documents qui permettent une vérification.

Par exemple, un paiement doit être corrélé avec une date, un numéro de chèque, et un numéro de facture au minimum. De

même, une liste de commissions doit être corrélée avec les transactions qui étaient à l'origine des commissions.

En d'autres termes, il faut lister les références croisées permettant une vérification complète des données.

### Digits de contrôle

Dans les cas où l'on utilise des codes longs, comme les inventaires, il est essentiel de faire appel à un codage redondant ou à des chiffres de contrôle.

Un *codage redondant* fournira des codes tels que :

TUYAU-204211
ECROU-418182
ASSBL-881921
FORME-329137

Les lettres de début du code identifient le produit pour un opérateur humain. Chaque « tuyau » a un code « 204 ». Chaque « ECROU » a un code « 418 ». Le programme vérifie que le code est en accord avec les lettres. Les erreurs n'arriveront au pire qu'entre les différents « tuyaux » ou « écrous » possibles, donc dans les trois derniers chiffres.

Cette méthode est facile à réaliser, mais fait perdre un peu de place.

Un *chiffre de contrôle* est un chiffre ajouté au code et qui détecte une transposition ou une erreur sur un seul chiffre.

Par exemple, dans 881921-5, le chiffre de contrôle est le 5. Il est calculé en multipliant 1 par n, 2 par $n^2$, 9 par $n^3$, etc., puis en ajoutant les 6 nombres. La somme est divisée par n. Si le reste est R, le chiffre de contrôle est n-R. Le choix de n est variable. Cette méthode est presque toujours utilisée pour vérifier les numéros de comptes bancaires, vu que l'erreur la plus fréquente dans ce cas, est une transposition de 2 chiffres, par exemple, 21224138 au lieu de 21221438.

### Sécurité des données

Les fichiers doivent être protégés tant en cas de mauvais fonctionnement que d'erreur ou de malveillance humaine.

Naturellement, tous les fichiers importants doivent être dupliqués à intervalles réguliers et sauvegardés.

De façon tout aussi essentielle, il faut dans un bon système de

fichiers disposer de sécurités software, telles que des mots de passe ou d'autres mécanismes de protection de l'accès.

Un mot de passe peut être imposé pour certains accès spécifiques à un fichier tels que lecture et écriture. Il ne doit pas être imprimé. Il ne doit pas être facilement accessible à l'intérieur du système. Une fois créé, un fichier peut ainsi être protégé contre les accès non autorisés ou contre une modification par une personne non qualifiée.

Les possibilités de *protection des fichiers* offerts par un bon système de fichiers comprennent la possibilité de spécifier des attributs d'accessibilité associés à un fichier comme la « protection écriture ».

Enfin, il ne faut pas sous-estimer les méthodes simples telles que un verrou à serrure sur le terminal.

**Autres précautions standards**

— A la fin de chaque journée, créer une copie de secours sur disque, bande ou papier afin que tout fichier endommagé ou perdu puisse être recréé.

— Protégez-vous contre les fluctuations du secteur si le problème affecte votre région. En particulier, le système devrait être physiquement débranché quand il n'est pas utilisé.

**Le choc ordinateur**

Les effets de l'informatisation peuvent être graves. Généralement, à moins que le personnel n'ait été formé en vue du passage à l'informatique, des délais considérables peuvent se produire dans le traitement, au début, entraînant un fonctionnement dégradé pendant un certain temps. Toutefois, on passe normalement assez vite à une plus grande efficacité à condition que des procédures correctes soient mises en place.

## RECAPITULATION

Il faut s'attendre à des malfonctions limitées du matériel et du logiciel. A condition que les précautions qu'on a indiquées soient prises, elles ne devraient pas affecter de manière importante le fonctionnement du système.

Le plus grand risque réside habituellement au niveau des

procédures (tant de gestion que de programmes) utilisées avec le nouveau système. Le bon sens, un bon jugement commercial, et les conseils d'un utilisateur expérimenté sont les moyens fondamentaux d'une réussite.

## ACHAT D'UN SYSTEME - RECAPITULATION

**1. Il doit y avoir un besoin :**

La nature et la quantité du travail doivent justifier le passage à l'informatique. Cela peut être mesuré au nombre de transactions ou de rapports identiques à générer. En outre, l'ordinateur peut être justifié par les problèmes particuliers qu'il peut résoudre dans des situations spécifiques comme la gestion d'inventaires, les fichiers d'adresses, les rapports pour la direction.

**2. Quand acheter ?**

Les prix de l'électronique diminuent constamment d'année en année tandis que les prix des périphériques tendent à rester stables ou à diminuer lentement.

Le système de demain sera toujours moins cher que celui d'aujourd'hui.

Mais l'introduction d'un ordinateur signifie une économie immédiate de N francs par mois. Retarder son installation de m mois équivaut à perdre m × N francs.

**3. Lequel acheter ?**

Les principales options et matériels disponibles ont été présentés. Le choix final vous appartient.

# CHAPITRE XIII
# AU SECOURS

## COMMENT OBTENIR DES INFORMATIONS

Il existe maintenant de nombreuses sources d'information qui facilitent la prise de décision :

— Publications.
— Clubs.
— Boutiques d'ordinateurs.
— Consultants.
— Instituts de formation ou firmes spécialisées dans la formation.

Nous allons passer en revue les principales.

## REVUES

La plupart des revues d'électronique ou d'informatique traditionnelles présentent régulièrement des articles sur les microordinateurs. Mais un certain nombre de nouvelles publications sont apparues depuis 1976 aux Etats-Unis, spécialement conçues pour l'utilisateur ou le constructeur de microordinateurs. Il est difficile de leur assigner un domaine déterminé vu que leur contenu évolue rapidement.

On peut toutefois dire que D$^{r}$ Dobb's Journal et Creative Computing comportent une section software substantielle dans tous les numéros, tandis que Interface Age, Byte et Kilobaud publient aussi des articles de hardware.

La plupart de ces revues sont en vente dans les boutiques microordinateurs.

## CLUBS ET BOUTIQUES

Un grand nombre de clubs d'amateurs ont fait leur apparition à travers les U.S.A. et en Europe et leurs adresses sont publiées

régulièrement dans les revues ci-dessus, en particulier dans les publications de People's Computer.

En France, l'AFIN a créé une section de constructeurs-amateurs utilisateurs, l'AFIN-CAU, 54, rue St-Lazare, 75009 Paris. On peut s'adresser aussi à COMPUTER CLUB EUROPE (même adresse que le journal JOCEN).

Les boutiques d'ordinateurs marquent un phénomène nouveau. La première a probablement créé à Santa Monica par Dick Heiser en été 1975 : Arrowhead Computer Company. Peu après, Paul Terrell ouvrit la première Byte Shop à Mountain View (Californie) : il est à l'origine de toute une chaîne de boutiques. L'une des plus importantes aux Etats-Unis est maintenant la chaîne des Computer-land.

Des centaines de magasins se sont ouverts dans chaque grande ville des Etat-Unis et d'Europe, qui ne vendent que des systèmes microordinateurs et produits associés à des amateurs ou à des hommes d'affaires. Les boutiques rendent des services importants : exposition de systèmes, maintenance et réparation, ensembles spéciaux (tant hardware que software), librairie spécialisée (livres et revues), conseils compétents, cours, bulletins, vente de logiciel.

La vente d'ordinateurs généraux est réputée exiger des qualités qu'on ne trouve pas chez les autres revendeurs de sorte que beaucoup de ces boutiques spécialisées prospèrent. Même IBM et DEC sont en train d'ouvrir des boutiques spécialisées.

Une bonne boutique permet de faire des comparaisons avant d'acheter et offre une assistance locale pour ceux qui démarrent.

Toutefois, la survie à long terme de beaucoup de ces boutiques n'est pas assurée car les microordinateurs deviennent disponibles dans les grands magasins et ailleurs.

## CONSULTANTS

Les consultants ont toujours été très utilisés pour les grosses installations et pour les moyennes. Aujourd'hui, le consultant *gratuit* est le patron ou le vendeur compétent d'une boutique microordinateur.

Les consultants traditionnels sont tout simplement trop chers proportionnellement, pour les utilisateurs de microordinateurs.

## FORMATION

La formation est toujours le meilleur investissement avant, ou s'il le faut, après, l'achat d'un système. Tout un choix de moyens sont offerts. Des cours sont souvent organisés par les boutiques, ou les universités ou par des firmes réputées. Des programmes sur vidéocassettes sont disponibles pour des groupes. Enfin, la lecture de livres et l'étude personnelle sont presque obligatoires pour un choix efficace et une bonne compréhension. (SYBEX est l'une des principales sociétés de formation en ce domaine).

## EXPOSITIONS

L'exposition qui a tout démarré est probablement l'exposition des ordinateurs personnels organisée à Atlantic City, en août 1976, par John Dilks.

Organisée dans la tradition des congrès de radio-amateurs, on prévoyait qu'elle attirerait peu de visiteurs. Il en est venu des milliers et les observateurs industriels se rendirent compte d'un nouveau phénomène : l'informatique personnelle était une réalité.

Le succès de cette exposition déclencha une vague d'expositions analogues à travers les Etats-Unis. Une des plus réussies sur la côte ouest a été la West Cost Computer Faire, organisée par Jim Warren (15 000 participants en 1978), devenue depuis une manifestation régulière.

Les expositions fournissent une occasion unique de voir d'un seul coup presque tous les matériels disponibles et d'obtenir commodément des informations et de la documentation. Elles offrent une bonne occasion de se former et leurs dates sont publiées régulièrement dans les revues de microordinateurs.

En France, la principale exposition organisée dans ce domaine est MICRO/EXPO qui se tient annuellement (organisée par notre société).

## HORS DES U.S.A.

Il existe maintenant à travers le monde, nombre de boutiques et de revues, mais moins qu'aux U.S.A. Une référence utile en Europe est Euromicro, l'Association Européenne pour la Microinformatique (adresse donnée avec les autres publications).

Des expositions sont aussi organisées dans les principaux pays, sur une échelle comparable à celle des Etat-Unis.

# REVUES SUR LES MICROORDINATEURS

**BYTE**
70 Main Street
Peterborough
New Hampshire 03458

**CALCULATORS and COMPUTERS**
DYMAX, P.O. Box 310
Menlo Park
Ca. 94025

**CREATIVE COMPUTING**
P.O. Box 789-M
Morristown, NJ

**D^r DOBBS JOURNAL**
Box 310
Menlo Park
CA. 94025

**EUROMICRO JOURNAL**
313, rue Lecourbe, 75015 Paris
France

**INTERFACE AGE**
Box 1234
Cerritos
CA. 90701

**JOCEN (Journal of Computer Encephalogia and Neurosis)**
Computer Club Europe
Postfach 11 08 68
D 6100 Darmstadt

**MICROPROCESSORS**
IPC Science and Technology Press Ltd.
IPC House, 32 High Street
Guilford GB GU1 3EW

**MICROSCOPE**
PO Box 141
CH 1000 Lausanne

**KILOBAUD**
1001001, INC.
Peterborough
New Hampshire 03458

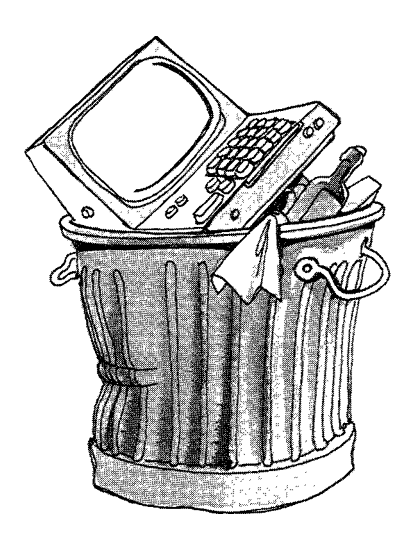
**PEOPLE'S COMPUTERS**
1263 El Camino Real, Box 5B
Menlo Park
CA. 94025

**PERSONAL COMPUTING**
401 Louisana S.E.
Albuquerque
New Mexico 87108

**SCCS INTERFACE**
P.O. Box 5429
Santa Monica
CA. 90405

**En France :**
**MICROSYSTEMES**
2-12, rue de Bellevue
75019 Paris

**MINIS ET MICROS/L'ORDINATEUR INDIVIDUEL**
41, rue de la Grange-aux-Belles
75015 Paris



INTRODUCTION AUX MICROORDINAT

# CHAPITRE XIV
# DEMAIN

## RECAPITULATION

Tout au long de ce volume, les principaux concepts relatifs aux microordinateurs ont été introduits. Des informations techniques supplémentaires peuvent être trouvées en appendice.

Le fonctionnement d'un système complet devrait maintenant être clair ainsi que les différents compromis qui interviennent dans le choix d'un équipement.

Le lecteur devrait avoir acquis toutes les connaissances nécessaires pour effectuer un choix rationnel. L'étape suivante est d'en apprendre plus et différents volumes complètent celui-ci dans le domaine du hardware (C4, C5) ou du software (C202,C3).

Cependant, la prochaine étape pourrait être aussi l'achat d'un système en vue d'une pratique effective (et aussi pour le plaisir). L'utilisation d'un ordinateur a été qualifiée de « jeu ultime » en ce sens qu'elle représente une extension de l'intelligence d'une personne, et la plupart des utilisateurs sont vite fascinés.

Mais ce domaine évolue rapidement. Que se prépare-t-il ?

## L'AVENIR

### Prix et miniaturisation

Avec le coût de l'électronique qui baisse continuellement et la miniaturisation qui progresse toujours, il est probable que la plupart des composants mécaniques ou électromécaniques des systèmes actuels vont devenir électroniques : il y correspondra une diminution du prix et de l'encombrement pour les claviers électroniques, les afficheurs électroniques, les mémoires de masse électroniques. Des microordinateurs en un seul boîtier sont déjà disponibles avec plusieurs milliers d'octets de mémoire sur le boîtier. Lorsque

certaines cartes seront standardisées, elles seront candidates à la réalisation en un seul boîtier.

Un système complet avec les fonctions qu'il remplit aujourd'hui, pourrait dans ce cas être réalisé dans le volume d'une calculatrice de poche, pour un prix analogue. Le problème principal reste celui des entrées-sorties à l'échelle humaine et il restera probablement le plus important facteur du prix.

La fonction de calcul elle-même est devenue pratiquement gratuite, de sorte qu'on peut maintenant introduire d'innombrables applications nouvelles. Voulez-vous en développer une ?

## CONCLUSION

Qui aurait prédit en 1975 la situation de ce domaine aujourd'hui ? Bien que de nombreuses spéculations aient été présentées, la réalité a dépassé toutes les prévisions. C'est le début d'une ère nouvelle, peut-être la « seconde révolution industrielle ». Personne ne peut faire de prédiction, même à court terme, vu la croissance explosive de cette industrie nouvelle.

Même si tout le monde n'utilisera pas un microordinateur, la vie de tous sera affectée par leur utilisation. Un de vos atouts les plus importants demain sera le même qu'aujourd'hui : la connaissance. Il est espéré que ce livre vous aura fait accomplir un pas sur ce chemin.

# APPENDICE A
# LA LOGIQUE DES ORDINATEURS

## INTRODUCTION

Voici une brève introduction aux concepts de base et aux symboles utilisés pour les circuits digitaux dans les microordinateurs.

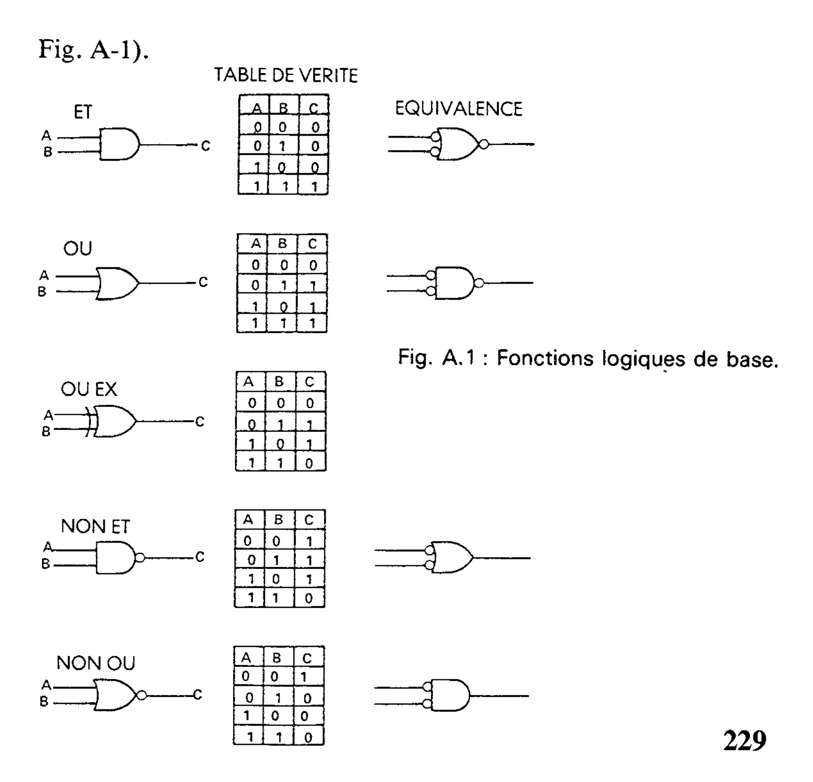
Toute information est représentée sous forme binaire et elle est traitée, modifiée ou transmise par de tels circuits.

## CIRCUITS LOGIQUES DE BASE

Les circuits logiques de base sont le ET, le OU, le NON-OU ainsi que les variantes OU exclusif, NON-ET et NON. Ils sont combinés pour produire toutes les autres fonctions.

La fonction de chaque « porte » est représentée par une table de vérité (Cf. Fig. A-1).

Fig. A-1).




ET

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

OU

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

OU EX

| A | B | C |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

NON ET

| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

NON OU

| A | B | C |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Fig. A.1 : Fonctions logiques de base.

Par exemple, la sortie C d'une porte ET sera à « 1 » si et seulement si à la fois A et B sont à « 1 ». Sinon elle sera à « 0 ».

De même, la sortie C d'une porte OU sera à « 1 » dès que l'une des entrées A ou B est à « 1 ».

**Bascules**

On a aussi besoin de circuits doués de mémoire qui emmagasinent un état « 0 » ou « 1 » et en changent lorsqu'on leur demande.

Les différentes bascules apparaissent sur les fig. A-2 et A-3.

— La bascule D introduit un délai, d'où son nom. Les sorties sont toujours complémentaires.

— La bascule J-K varie sa réponse en fonction de la combinaison des entrées (voir table de vérité). La sortie Q est « synchronisée » : elle ne change qu'après l'impulsion d'horloge.

— La bascule R-S (set-reset) est la plus simple. Elle forme un verrou asynchrone ou synchrone (si on utilise une horloge).

Les réalisations possibles de bascules à l'aide de portes élémentaires apparaissent fig. A-4.

Un *registre* est un ensemble de bascules (8 pour un octet).

Enfin, divers symboles électriques ou électroniques apparaissent fig. A-5 et A-6.

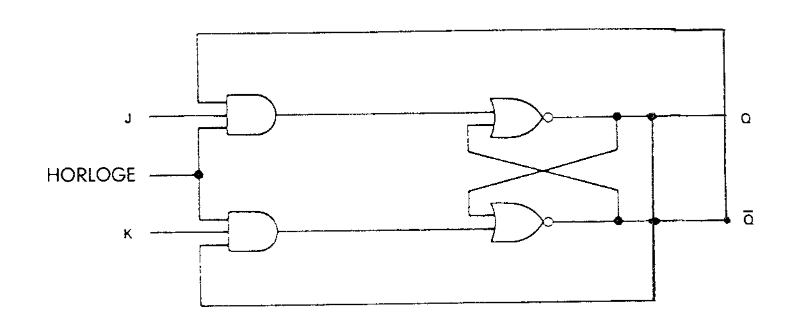



Fig. A-2: Bascule JK.

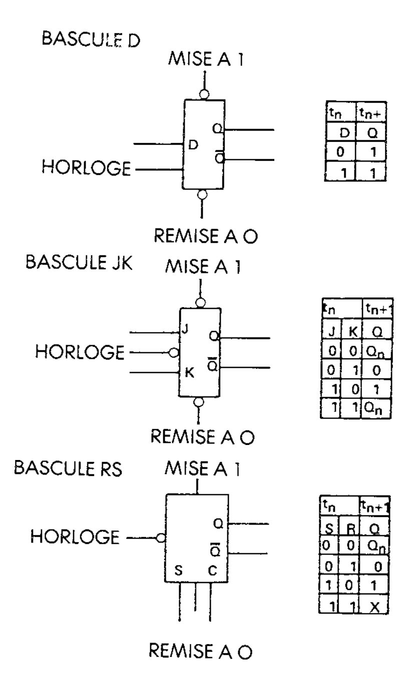
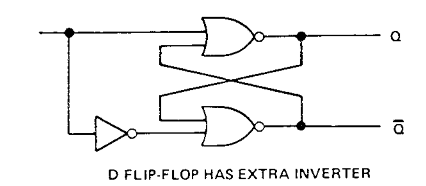



Fig. A-3 : Bascules.



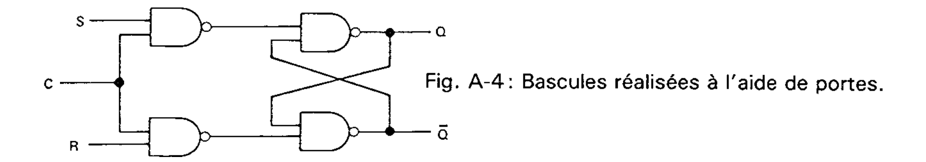




Fig. A-4 : Bascules réalisées à l'aide de portes.

TERRE

MASSE

CONNEXION

PAS DE CONTACT

QUARTZ

LUMIERE

RESISTANCE

RHEOSTAT (RESISTANCE VARIABLE)

BOBINE

DIODE

LED

DIODE ZENER

TRANSISTOR NPN

TRANSISTOR PNP

INTERRUPTEUR

INVERSEUR

BOUTON POUSSOIR

TRANSFORMATEUR

BATTERIE

Fig. A-5 : Symboles.

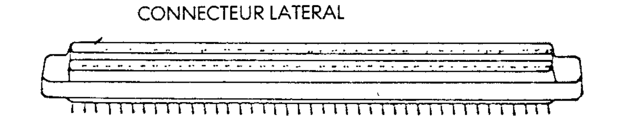
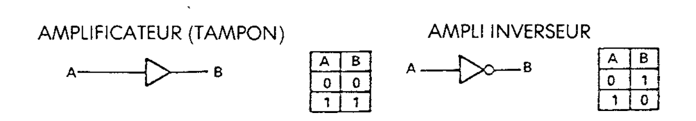



Fig. A-6 : Connecteur pour circuit imprimé.




| A | B |
|---|---|
| 0 | 0 |
| 1 | 1 |

| A | B |
|---|---|
| 0 | 1 |
| 1 | 0 |

Fig. A-7 : Amplificateur.

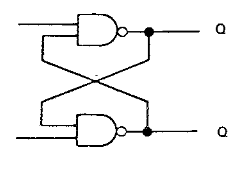



Fig. A-8 : Verrou.

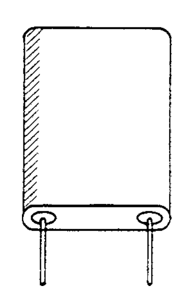
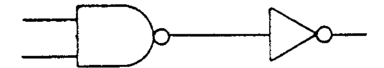


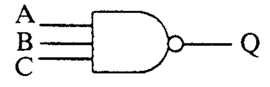
Fig. A-9 : Quartz.

## EXERCICES

Quelle est la fonction de la porte réalisée par le circuit suivant ?



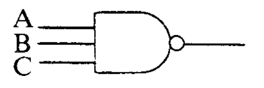
*Réponse :* ET
Construire la table de vérité d'une porte ET à 3 entrées.




*Réponse :*

| A | B | C | Q |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Construire la table de vérité d'une porte NON-ET à 3 entrées.




*Réponse :*

| A | B | C | Q |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

# APPENDICE B
# BITS ET OCTETS

## LE SYSTEME BINAIRE

Ce chapitre est une brève introduction au système binaire qui est utilisé pour représenter les informations à l'intérieur de l'ordinateur. La représentation des nombres est en principe facile. Voyons quelques exemples :

« 3 » a pour représentation « 11 »

« 5 » a pour représentation « 101 »

Le système binaire utilise des « 0 » et des « 1 » pour représenter tous les nombres. L'équivalent décimal est calculé comme suit :

11 est $1 \times 2 + 1 \times 1 = 3$

101 est $1 \times 4 + 0 \times 2 + 1 \times 1 = 5$

Le « bit » (chiffre binaire) le plus à droite représente $2^0 = 1$» On l'appelle « bit le moins significatif » (LSB). Quand on va de droite à gauche, comme dans le système décimal, chaque bit successif représente la puissance de 2 correspondante. Ainsi, dans 101, le bit le plus à droite a pour poids 0, le suivant $2^1 = 2$, le suivant $2^2 = 4$.

Nous savons maintenant calculer l'équivalent décimal : si $b_n$ $b_{n-1} \ldots b_1 b_0$ est un nombre binaire, son équivalent décimal est :
$b_n x 2^n + b_{n-1} x 2^{n-1} + \ldots + b_1 x 2 + b_0$ où les b sont des « 0 » ou des « 1 ».

## DE DECIMAL A BINAIRE

La conversion inverse est tout aussi simple :

16 divisé par 2 = 8 reste **0**

8 divisé par 2 = 4 reste **0**

4 divisé par 2 = 2 reste **0**

2 divisé par 2 = **1** reste **0**

Le dernier quotient et les restes successifs forment de bas en haut : 10000. C'est 16 en binaire.

Autre exemple :
11 divisé par 2 = 5 reste **1**
5 divisé par 2 = 2 reste **1**
2 divisé par 2 = **1** reste **0**
La réponse est 1011.

**EXERCICE :** *Convertir 1000 et 1011 en décimal afin de vérifier les conversions précédentes.*

## REPRESENTATION DES NOMBRES

Nous venons de montrer que les nombres décimaux peuvent être représentés par des bits. En pratique, les microordinateurs courants structurent toute information en groupes de 8 bits ou octets.

**EXERCICE :** *Quel est l'entier le plus grand qui peut être représenté par un octet ?*
*REPONSE :* $2^8 = 256$

**EXERCICE :** *Quel est l'entier le plus grand qui peut tenir en deux octets ?*
*REPONSE :* $2^{16} = 65\,536 = 64$ K

### Arithmétique binaire

Les opérations arithmétiques sont effectuées de la même manière que dans le système décimal :

```
    0101    (5)
+   0110    (6)
--------
=   1011    (11)
```

Les règles de l'addition sont :
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10 (ou 0 avec une retenue de 1)

### Nombres négatifs et fractionnaires

Le problème suivant est de représenter les nombres négatifs et fractionnaires. Ceci est traité dans nos autres livres (cf. C4 ou C202).

Disons brièvement qu'on emploie généralement pour les nombres avec signe la notation en « complément à 2 » tandis que les nombres fractionnaires requièrent une représentation en « virgule flottante ».

## CARACTERES

Les caractères sont (presque) universellement codés en ASCII sur 8 bits. La table de conversion ASCII apparaît fig. 5-.2.

Sept bits permettent de coder $2^7 = 128$ caractères différents. Cela suffit pour :

— Les 26 lettres de l'alphabet (A à Z) majuscules.
— Les 26 lettres (a à z) minuscules.
— Les 10 chiffres décimaux (0 à 9).
— 66 caractères spéciaux.

On n'a presque jamais besoin de plus de caractères, de sorte que ce code est universellement utilisé. En pratique, plusieurs « caractères » sont des caractères de *contrôle :* ils ne s'impriment pas mais servent de commandes ou informations d'état entre l'ordinateur et le terminal. Par exemple ESC (escape) sert généralement de commande d'arrêt.

**Question :** *A quoi sert le 8^e^ bit ?*
**Réponse :** C'est le bit de *parité.*

Illustrons maintenant l'usage du binaire à l'aide d'un dernier exemple.

L'utilisateur tape « JE DIS 12 ».

Le moniteur ou l'éditeur va déceler ces caractères au clavier et les ranger en mémoire sous forme binaire.

Supposons que le code pour
espace soit 10100000
J soit 11001010
E soit 11000101
D soit 01000100
I soit 11001001
S soit 01010011
1 soit 10110001
2 soit 10110010

La représentation de « JE DIS 12 » est :
11001010
11000101

10100000
01000100
11001001
01010011
10100000
10110001
10110010

**Question :** *N'y-a-t-il pas une erreur dans la représentation de « 12 » ?*
**Réponse :** Non. Quand les chiffres apparaissent dans une chaîne de caractères, ils sont traités comme *des caractères ASCII ordinaires :* « 12 » est représenté par le caractère ASCII « 1 » suivi du caractère ASCII « 2 ». Un éditeur ne sait rien au sujet des nombres. Il ne manipule que des caractères. Si l'on utilise un « assembleur », il devra « savoir » qu'un nombre est un nombre, généralement en le trouvant à la droite d'un opérateur comme =, + ou —.

Alors il le codera en binaire (généralement complément à deux), afin de pouvoir facilement effectuer des opérations arithmétiques sur lui.

**Question pour le lecteur attentif :** *Qui fait la conversion de la touche enfoncée sur le clavier à l'octet qui représente la caractère : 1 - le microprocesseur, 2 - le codeur de clavier ?*
**Réponse :** Si le clavier à un codeur (les claviers alphanumériques en ont un généralement) alors c'est celui-ci. Sinon (cas, en général, des petits claviers 16 ou 24 touches), c'est un programme de conversion (qui fait partie du « moniteur ») qui effectue le codage.

# APPENDICE C
# NOTIONS FONDAMENTALES SUR LES COMMUNICATIONS AVEC L'ORDINATEUR

## PARALLELE OU SERIE

Il y a quatre options principales pour les échanges d'information entre un ordinateur et ses périphériques : parallèle ou série, synchrone ou asynchrone.

### En parallèle

Pour un microordinateur standard, la transmission parallèle fait intervenir 8 bits donc un bus de 8 fils. Bien entendu, la transmission parallèle est plus rapide que la transmission série dans laquelle les bits sont transmis l'un après l'autre.

Partout où la vitesse est un impératif, il faut faire appel à une transmission parallèle.

C'est toujours le cas *à l'intérieur* de l'unité centrale où les cartes doivent échanger des informations à la plus grande vitesse possible. Le bus qui parcourt le *fond de panier* et transporte les signaux d'une carte à l'autre est, par suite, toujours parallèle. C'est le cas, par exemple, du bus S100, déjà décrit.

Hors de l'ordinateur, un bus parallèle nécessite 8 lignes pour les données plus un grand nombre de lignes de contrôle pour les signaux de synchronisation. Il sera nécessaire par exemple pour une imprimante rapide.

### En série

La transmission série consiste à envoyer les bits sur une seule ligne, l'un après l'autre. Pour séparer deux bits successifs, il faut en

outre un signal d'horloge. L'avantage évident est le bas prix de la ligne de communication (rien que 2 fils). Mais en raison de la bande passante limitée afin que la transmission reste fiable, cette technique limite la vitesse de transmission. D'autre part, elle nécessite un tampon de 8-bits à chaque extrémité pour emmagasiner un caractère.

Cependant, elle est simple et économique et elle est utilisée pour la plupart des « périphériques lents » comme les Télétypes ou les terminaux à écran.

Les vitesses utilisables varient de 110 baud à 9 600 baud. Dans le monde de l'informatique, le baud (en l'honneur de Baudot) représente un bit par seconde (bps).

Le Télétype fonctionne à 110 baud (10 caractères par seconde, chaque caractère comportant 11 bits).

Le Decwriter LA 36 va à 300 baud. Les terminaux à écran standards vont jusqu'à 9 600 baud.

### Interfaces série

Les deux interfaces série standards les plus utilisés sont le RS232C et la boucle de courant 20 mA. La boucle 20 mA est l'interface habituel des Télétypes, où un « 1 » s'appelle « marque » (le courant passe) et un « 0 » s'appelle « espace » (courant interrompu).

Le RS232C sert aux écrans et aux liaisons téléphoniques. Le « 1 » est représenté par une tension de + 3 à + 15 V, le « 0 » par – 3 à – 15 V.

## SYNCHRONE OU ASYNCHRONE

L'information peut être envoyée octet par octet : c'est une transmission asynchrone. Le terminal ne « sait » jamais quand un caractère va arriver. Pour reconnaître un caractère, il faut transporter une certaine information de synchronisation en même temps que le caractère.

Ainsi le Télétype utilise un code ASCII à 8 bits pour le caractère, précédé d'un bit de START (départ) et de deux bits d'arrêt (STOP). C'est pourquoi un Télétype 10 cps (caractères par seconde) doit fonctionner à 110 baud : chaque caractère comporte 11 bits.

Les communications asynchrones sont utilisées chaque fois que c'est possible car elles sont simples et fiables.

En transmission synchrone, on envoie des « paquets » ou des

blocs de données, qui doivent être strictement synchronisés. Elle permet des vitesses potentiellement plus élevées, mais nécessite beaucoup plus de logique et est donc plus chère. Elle peut être utilisée par exemple pour des communications rapides avec un autre ordinateur.

## AU-DELA DU SYSTEME MICROORDINATEUR

Le microordinateur peut communiquer avec des périphériques (ou d'autres microordinateurs) à distance par l'entremise de lignes téléphoniques (ou, si la distance est courte, à l'aide d'une paire torsadée). Pour transmettre de l'information à longue distance, de façon fiable, sur une ligne téléphonique, il faut la traduire en fréquences audibles (sous forme série). Cela est accompli par un *modem*.

Le modem (« modulateur-démodulateur ») transforme les données binaires série en fréquences et il décode les fréquences en bits série. La technique la plus utilisée est le FSK (frequency shift keying = codage par changement de fréquence). L'état « porteuse » ou « pas de donnée » est représenté par une certaine fréquence. Un « 1 » est créé en envoyant une fréquence plus élevée (« décalage » vers le haut). Un « 0 » est créé en envoyant une fréquence plus basse.

Une fréquence centrale peut être utilisée pour l'émission et une autre, bien distincte de la première, pour la réception. Une liaison où les données peuvent voyager dans les deux sens s'appelle « duplex complet ». Sinon c'est du semi-duplex.

Pour des grandes vitesses de transmission, on peut louer des lignes téléphoniques de qualité supérieure. C'est ainsi que les gros ordinateurs sont reliés pour former des réseaux.

## RESUME

A l'intérieur du coffret microordinateur, tous les transferts d'informations sont normalement parallèles.

Au dehors, les transferts se font en série par un connecteur RS232 ou une boucle de courant 20mA vers les périphériques lents, et en parallèle par des connecteurs et des ports spécialisés vers les périphériques rapides.

Enfin, le microordinateur peut même communiquer par téléphone, à l'aide d'un modem.

# APPENDICE D
# FICHIERS
# ET ENREGISTREMENTS

## POURQUOI ?

Du point de vue de l'utilisateur, une entité d'informations s'appelle un *fichier*. Un fichier peut être un programme, une liste d'adresses, une liste de comptes à recevoir.

Il est commode que l'utilisateur puisse attribuer un nom à chaque fichier qu'il utilise. Par exemple il dira :

« CHARGER PAIE (DEPUIS LE DISQUE) »

Néanmoins les fichiers peuvent être très divers et ils posent de nombreux problèmes pratiques :

— un fichier peut avoir n'importe quelle longueur

— il peut s'étendre ou se réduire

— on peut avoir besoin d'accéder rapidement aux informations situées au milieu du fichier

— il faut disposer de certaines informations au sujet du fichier : quelle est sa longueur ? est-il en binaire ou en ASCII ?

— son contenu doit être exact et en sécurité.

Afin de résoudre ces problèmes, des techniques se sont développées ; il en résulte tout un choix d'options pour « structurer » les fichiers. Passons en revue les principales.

## ALLOCATION D'ESPACE

Tout fichier peut normalement varier en longueur. Un fichier « personnel », par exemple, doit normalement grossir, et parfois diminuer.

Or tout support de stockage, disque ou bande, est fini. La méthode la plus facile est d'allouer une disquette entière, ou toute une bande, au fichier : il aura toute la place pour s'étendre.

Malheureusement, il n'y a habituellement, à un instant donné, dans un système qu'une (ou deux) bandes ou disquettes. La solution précédente « gaspille de la place », c'est-à-dire qu'elle restreint le nombre de fichiers auxquels l'utilisateur aura accès simultanément (s'il n'y a q'un fichier par unité).

Une amélioration évidente est de définir une longueur maximale pour le fichier, disons une demie cassette. De cette façon, il y aura deux fichiers par cassette.

Si le fichier est très petit au départ, il est probable que cette solution va, là encore, gaspiller les 3/4 de l'espace disponible.

### Fichiers séquentiels

Du point de vue de l'utilisation de l'espace, le mieux serait de ranger des fichiers « en séquence », c'est-à-dire à la suite les uns des autres.

Malheureusement, si l'un des fichiers s'étend, alors tous les autres doivent être déplacés, ou alors le suivant doit être supprimé. C'est un procédé long et fastidieux.

C'est ce qui est fait dans la plupart des systèmes à bandes. En pratique, au lieu de déplacer un fichier, on utilise le reste de l'espace libre sur la cassette, étant donné que les cassettes ont une grande capacité.

L'autre inconvénient est le problème de *l'accès :* l'accès est, par essence, séquentiel. Il est difficile de retrouver des informations dispersées de façon aléatoire.



Fig. D.1 : Fichier séquentiel.

### Enregistrements et blocs

Pour faciliter l'allocation d'espace à un fichier, la plupart des « systèmes de fichiers » structurent l'espace disponible en *blocs*. Ces blocs peuvent être de taille fixe ou variable. Dans la plupart des cas ils sont de taille fixe, pour raison de simplicité. Nous allons les supposer tous égaux. On les appellera, selon le système, « secteurs », « pages », « enregistrements ».

Un fichier va donc maintenant occuper plusieurs blocs, dont le dernier ne sera généralement pas plein.

Avec cette technique il n'y a en moyenne qu'un demi-bloc de perdu. Comment le fichier s'étend-il ? Très simplement : un ou plusieurs blocs sont alloués au fichier s'ils sont disponibles.

Le problème suivant est de garder trace des blocs alloués au fichier, vu qu'ils ne sont probablement pas en séquence.

### Structuration d'un fichier

On peut utiliser deux structures fondamentales : la table des matières ou une structure de liste.

**La table des matières** (ou répertoire, ou index)

Dans une structure avec table des matières, un « bloc-répertoire » (ou un fichier-répertoire) contient les *« pointeurs »* (adresses) de tous les blocs qui appartiennent au fichier.

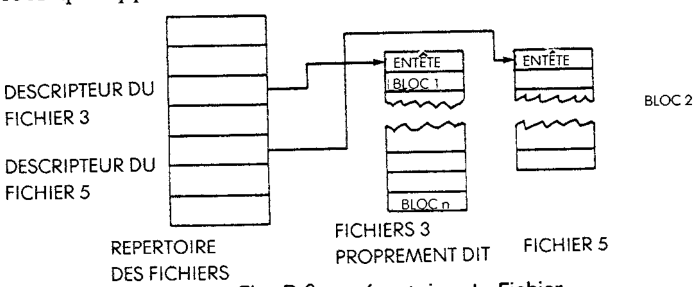



Fig. D.2 : répertoire de Fichier

Par exemple, sur une disquette, le répertoire concernant le fichier « paie » contiendra :

1 - 2
1 - 3
1 - 4
2 - 5

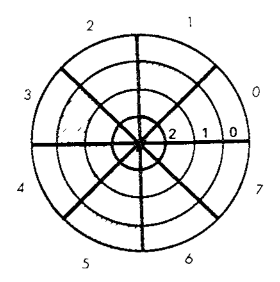
ce qui peut être lu sous la forme :

le premier bloc est le secteur 2 de la piste 1
le deuxième bloc est le secteur 3 de la piste 1
le troisième bloc est le secteur 4 de la piste 1
le bloc suivant est le secteur 5 de la piste 2

C'est une méthode simple et facile. Mais, pour être efficace il faut que le répertoire soit lu et réside en mémoire centrale. A la fin de chaque bloc, il faut accéder au répertoire pour déterminer quel est le bloc suivant.

**Exercice :** *Comment le système de fichiers peut-il garder trace des blocs disponibles afin de pouvoir en allouer ou en libérer ?*

**Réponse :** Examiner la fig. D-3.




| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

Fig. D.3 : Une carte d'allocation, dont chaque élément est un bit, peut servir à la gestion des secteurs du disque.

## Structure de liste

Une autre solution est de conserver cette information dans chaque bloc. Ceci est illustré par la fig. D-4. Chaque bloc contient un « pointeur » vers le bloc suivant qui appartient au même fichier. Une marque spéciale, « EOF » (fin de fichier) indique la fin de la liste.

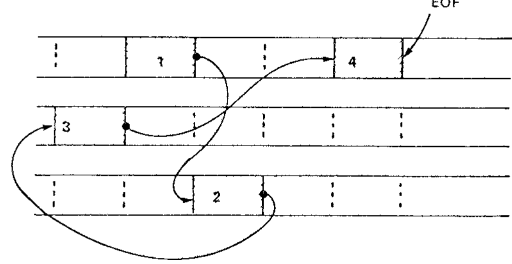



Fig. D.4 : Blocs structurés en liste.

## Le séquentiel-indexé

On peut faire appel à des méthodes hybrides, comme un répertoire (un « index ») de fichiers séquentiels. Ceci est appelé l'accès séquentiel indexé, et constitue une amélioration notable par rapport à l'accès séquentiel simple.

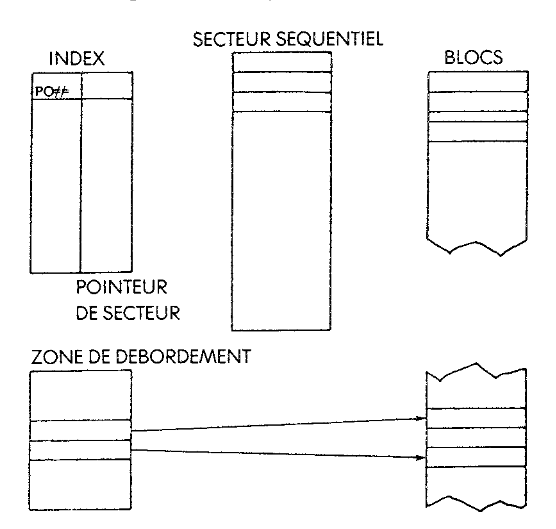



Fig. D.6 : Arbre des listings de gestion d'une firme.

## RECUPERATION DE L'INFORMATION

Le problème suivant à résoudre, est d'accéder efficacement aux informations. Dans un fichier séquentiel, il est nécessaire pour accéder à un élément de lire complètement tout ce qui le précède dans le fichier. C'est très lent.

Le problème est résolu par la structuration. Plus le mécanisme d'accès est efficient, plus il faut une structure élaborée. Généralement, la structuration entraîne des répertoires successifs ou des chaînages complexes. Il en résulte un temps d'accès *minimum* plus long mais un temps d'accès *moyen* plus court.

### Arborescences

Une des méthodes les plus simples pour structurer un fichier est de faire appel à une arborescence.

Un répertoire-maître pointe vers des sous-répertoires et ainsi de suite jusqu'au fichier lui-même.

Pour retrouver une information, il faut « cheminer dans l'arborescence ».

On peut utiliser d'autres structures pour stocker les informations et gérer les blocs du fichier proprement dit. Mais nous sortons du sujet de ce livre.

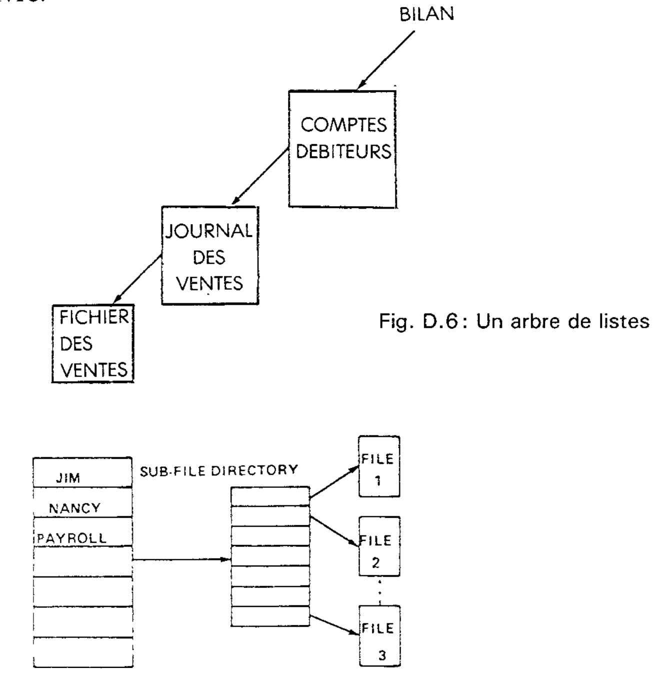



Fig. D.6 : Un arbre de listes

Fig. D.7 : Une arborescence de répertoire à deux niveaux.

# APPENDICE E

## QUELQUES CONSTRUCTEURS DE PETITS SYSTEMES DE GESTION

**Basic Four**
14101 Myfard Rd.
Tustin, CA 92680

**Burroughs**
Burroughs Pl.
Detroit, MI 48232

**Data General**
Route 9
Southboro, MA 01772

**Datapoint**
9725 Datapoint Dr.
San Antonio, TX 78284

**Digital Equipment Corporation**
Parker St.
Maynard, MA 01754

**Four Phase Systems**
19333 Vallco Parkway
Cupertino, CA 95015

**General Automation**
1055 S. East St.
Anaheim, CA 92805

**Hewlett Packard**
11000 Wolfe Rd.
Cupertino, CA 95014

**Honeywell**
200 Smith St.
Waltham MA 02154

**IBM**
Atlanta, GA 30301

**Logical Machine**
887A Milten Rd.
Burlingame, CA 94010

**Microdata**
17481 Red Hill Ave.
Irvine, CA 92705

**NCR**
Main and K St.
Dayton OH 45409

**Olivetti**
500 Park Ave.
New York, NY 10022

**Philips**
175 Froelich Farm Blvd.
Woodbury, NY 11797

**Quantel**
3525 Breakwater Ave.
Hayward CA 94545

**Sycor**
100 Phoenix Dr.
Ann Arbor, MI 48104

**Wang**
836 North St.
Tewksbury MA 08176

# APPENDICE F

## CONSTRUCTEURS DE MICROORDINATEURS

**Alpha micro Systems**
17875 N. Sky Park
North Irvine, CA 92714
(714) 957-1404

**Apple Computer, Inc.**
20863 Stevens Creek Blvd.
Cupertino, CA 95014

**Commodore**
901 California Ave.
Palo Alto, CA 94304

**Cromemco**
2400 Charleston Rd.
Mountain View, CA 94043
(415) 964-7400

**Data General**
Southboro, MA. 01722
(617) 485-9100
Telex : 48460

**Digital Group**
P.O. Box 6528
Denver, CO. 80206
(303) 777-7133

**E et L Instruments, Inc.**
61 First St.
Derby CT 06418
(203) 735-8774

**Equinox Division**
Parasitic Engineering
P.O. Box 6314
Albany, CA 94706

**Extensys Corp.**
592 Weddell Dr., S-3
Sunnyvale, CA 94086
(409) 734-1525

**Heath Co.**
Benton Harbor, MI 49022

**IMS Associates, Inc.**
(IMSAI)
14860 Wicks Blvd.
San Leandro, CA 94577
(415) 483-2093

**MITS (Altair)**
2450 Alamo S.E.
Albuquerque, NM 87106

**North Star**
2547 Ninth St.
Berkeley, CA 94710
(415) 549-0858

**Polymorphic Systems**
737 S. Kellogg
Galeta, CA 94608

**Processor Technology**
Box G, 7100 Johnson Industrial Drive
Pleasanton, CA 94566
(415) 829-2600

**SWTPC**
219 W. Rhapsody
San Antonio, TX 78216

**Technical Design Labs**
Research Park, Bldg. H
1101 State Road
Priceton, N.J. 08540
(609) 921-0321

# INDEX

A


| | |
|---|---|
| ABS | 82 |
| Accès aléatoire | 53 |
| Accès direct | 87 |
| Achat d'un système | 117 |
| ADC | 168 |
| Adresse | 36 |
| Affichage | 15 |
| Aide à la mise au point | 72 |
| Aiguilles | 155 |
| Howard Aiken | 171 |
| Algorithme | 61 |
| Alimentation | 31,58 |
| Alpha Micro | 202 |
| Alphanumérique | 37 |
| Altair | 173,186 |
| Alternatives a basic | 90 |
| ALU | 35 |
| Amplificateur | 232 |
| Amplificateur de bus «divers» | 29 |
| APL | 76,90 |
| APPLE II | 185 |
| Applications des microordinateurs | 27 |
| Applications individuelles | 27 |
| Arborescences | 246 |
| Arcade de Bally | 179 |
| Architecture | 47 |
| Arithmétique binaire | 69,236 |
| ASCII | 67,238 |
| ASR | 150 |
| Assembleur | 62,72 |
| Assignation | 87 |
| Asynchrone | 240 |
| Au secours | 221 |
| Audit trail | 105 |
| Avantages | 43 |
| Avenir | 227 |

B

Babbage 171
Banc d'essai 124
Bande magnétique 23,163
Basic 63,76,126,115
Basic compile 88
Basic étendu 89
Basic de gestion 86
Basic standard 89
Bascules 230
Baud 138
Bibliothèques de programmes 43,65
Bit 35
Bit d'état 50
«bit le moins significatif» (LSB) 235
Blocs 104,243
Boîtier 28
Boucles 64,81
Boucle de courant 138
Boucles imbriquées 81
Branchement 28,49
Broches 48
Bus 29,35,137
Bus adresse 49
Bus de commande 36
Bus de données 36,49
Byte 224
Byte Shope 222
Bytemaster de Digital Group 199

C

CALL 86
Calcul en flottant 86
Calcul scientifique 171
Calculateur 28
Capacité 109
Caractères 237
Carte 31
CBASIC 90
CCD 164
CDC 172
Chaînes de caractères 86
Challenger de Ohio Scientific 195
Charger 23

Chargeur 72
Chiffre de contrôle 218
Chiffres significatifs 85
Chip 28
choc ordinateur 219
Choix 171
Choix d'un système 121
Circuits électroniques 67
Circuit intégré 34
Circuits logiques 229
Clavier 15,18,29,32,37,55,141
Clavier alphanumérique 132
Clavier commercial 141
Clubs et boutiques 221
Codage redondant 218
Codes 67
Code client 104
Code-objet 63
Coffret 38,207
Commande 23
Commentaires 79
Commodité 128
Communications 239
Communication serie 55
Compilateur 63,88
Compilé 63
Composants 47
Comptes clients 98
Comptes débiteurs 105
Compteur 48
Compteur de programme 48
Compucolor II 200
Computer -land 222
Connecteur 138,232
Constructeurs 248
Consultants 222
Contrôles 217
Contrôleurs 31,58,38
Contrôle de processus 27
Contrôle de zone 217
Conversion 236
Couleur 146
Coût 207
Coûts invisibles 207
Coût minimal 133
CP/M 90,205

Crayon lumineux 164
CRC 215
Creative Computing 224
Critères 125
Critères de choix 121
CROMEMCO Z2 191
CRT 18,142
CRTC 58
Curseur 20,146

D

DAC 168
DATA 78
Debugger 72
DEC 172
Décodeur 49
Définitions 28
Demain 227
Déroulement 142
Développement 73
Développement d'un programme 71
Diffusion 34
Digital Equipment 198,200
Digits de contrôle 218
John Dilks 223
Diode émettrice de lumière (LED) 19
DIP 32
Disque 73,157
Disque double face 162
Disque souple « floppydisk » 32,161
Disquette 109,160
Documentation 128
Données 29,53
DOS 55,73
Double-densité 162
Dr. Dobbs 224

E

echecs 213
Echo 20
Eckart et Manchly 171
Ecran cathodique 142
Ecran de visualisation 29
Ecran T.V 38

Editeur 71,237
EDSAC 172
EDVAC 172
Encodeur 37
ENIAC 171
Enregistrements 243
Enregistreur à cassettes 18
Entrée 62
Entrée-sortie 29,37
Entrée vocale 166
Enseignement 127,132
environnement 213
Equinox de Parasitic Engineering 195
EPROM 54
Erreurs de frappe 24
Erreurs software 215
ET 69
EUROMICRO 224
Evolution des prix 122
exactitude des données 217
Exécutif 23
Exécution 49
Exposant 80
Expositions 223

F

Fabrication 32
Fiabilité 129,214
Fichiers 32,61,98,242
Fichier d'adresses 107
Fichier du personnel 97
Fichiers séquentiels 86,243
Firmware 29
Fond de panier 239
FOR 83
Formatage 86
Formation 223
Formats de sortie 87
FORTRAN 64,76
Fréquence d'horloge 123
FSK 241
Fusible 54

G

Galettes 34
Gestion 84,96,127
Gestion des commandes 98
Gestion des transactions 98
Gestion d'inventaire 208
GO TO 79

H

Hardware 96
Heathkit 197
Dick Heiser 222
Héxadécimal 69
Horizon de North Star 194
Horloge 35,57,123

I

IBM 161,171
IEEE 488 179
IF...THEN 81
Impact 149
Imprimante 37,148
Imprimante à chaîne 155
Imprimante à marguerite 151
Imprimante à ruban 149
Imprimante électrosensible 149
Imprimante-ligne 157
Imprimante matricielle 153
Imprimante thermique 148
Impuretés 34
Imsai 187
Incrémenteur 49
Informations scientifiques 64
INPUT 78
Instruction 28,48,49,61
Instructions arithmétiques 28
Instructions logiques 28
INT 82
Intel 90,173
Intensités 146
Interactif 89
Interface 19
Interface Age 224

| | |
|---|---|
| Interface parallèle | 138 |
| Interface standard | 138 |
| Interpreteur | 63 |
| Interrupteur | 167 |
| Inventaire | 98,106 |

J

| | |
|---|---|
| Jeu | 23 |
| Journal | 105 |
| Journal des ventes | 98 |
| Journaux | 98 |

K

| | |
|---|---|
| K | 37 |
| Kilobaud | 225 |
| KIM-1 | 174 |
| Kit | 130 |
| KSR | 150 |

L

| | |
|---|---|
| Langage de haut niveau | 63 |
| Langage de programmation | 64 |
| Langage lambda | 91 |
| Langage machine | 61,62 |
| Langage naturel | 93 |
| LEDS | 167 |
| Lequel acheter | 117 |
| LET | 78 |
| Lignes | 145 |
| Limitations du basic | 89 |
| Listes | 83 |
| Logiciel | 73,97,127 |
| Logiciels disponibles | 110 |
| Logiciel «Software» | 29 |
| Logique | 229 |
| LSI | 28 |
| LSI 11 | 198 |

M


| | |
|---|---|
| «Macy's» | 176 |
| Manche à balai | 165 |
| MARK I | 171 |
| Matériel complet | 126 |
| Matériel «Hardware» | 29 |
| Mémoire | 36,53 |
| Mémoires à bulles | 164 |
| Mémoire à long terme | 23 |
| Mémoire auxiliaire | 37 |
| Mémoire centrale | 29,37 |
| Mémoire de masse | 164 |
| Mémoires mortes «ROM» | 29,37 |
| Mémoires statiques | 53 |
| Mémoires vives | 37,53 |
| Menu | 101 |
| MICRO/EXPO | 223 |
| Micro-historique | 171 |
| Microordinateurs | 28,39 |
| Microordinateurs de gestion | 27 |
| microprocesseurs | 28,29,173,224 |
| Microprogrammes | 29 |
| Microscope | 225 |
| Microsoft | 90 |
| Mini-basic | 85,133 |
| Minidisquette | 161 |
| Minis et micros | 225 |
| Miniordinateur | 114 |
| Mise à jour | 107 |
| Mise à jour automatique | 107 |
| Mise en œuvre | 128 |
| MITS | 173 |
| Modem | 241 |
| Moniteur | 23,74,123,147,237 |
| Moniteur vidéo | 142 |
| MOORE | 172 |
| MOS | 16,53 |
| MOS Technology | 173 |
| Mot-clé | 78 |
| Mot de passe | 21 |
| Mots | 49 |
| Motorola | 173 |
| MPU | 47 |
| MSI | 28 |

N

Nanosecondes 48
NEC TK 80 176
Nombres négatifs 236
NON 69,229
Notation scientifique 80

O

Octal 70
Octet 36,67
Opérations logiques 69
Ordinateur 23,28
Organigrammes 65
OU 69,229

P

Packages 110,115
Panneau avant 31
Panneau de commande 136
Pannes électriques 214
Pannes Hardware 213
Pannes mécaniques 213
Parallèle 239
Parité 214,237
Pascal 90,171
Pastille 28
Paye 97
PC 48
PDP 11 200
PDP 11/03 198
People's Computing 225
Performance 64,128,123
Périphériques 32,141
P.E.T. Commodore 19,179
PIO 7,58
PL/I 90
PL/M 63,90
Points d'arrêt 72
Polymorphic 88 196
Port d'entrée/sortie 58
Portes 231

Précaution 117
Précautions standards 219
Prix 121
Prix et miniaturisation 227
procédures d'utilisation 216
procédures informatisées 208
Programme 28,61
Programme de traitement 100
Programme source 64
Programme transactionnel 97
PROM 54
Puce 29

Q

Quand acheter 117,209
Quartets 36
Quartz 35 , 233

R

RAM 37,53
RAM dynamiques 55
Registre 29,48,50,230
Registre d'état 50
Registre d'instruction 49
Registres-tampons 58
Relais 167
REM 79
Remise à zéro 20
Représentation 236
Représentation externe 70
Représentation de l'information 67
Reset 20
Retour chariot 21
Return 82
Revues 221,224
Rollover 141
ROM 53
RPROM 54
RS 232 19,39

S


| | |
|---|---|
| SCCS Interface | 225 |
| Secteurs | 243 |
| sécurité des données | 218 |
| Selectric | 148,151 |
| Séquentiel-indexé | 245 |
| Série | 138,239 |
| SGN | 82 |
| Shugart | 161 |
| Signaux de commande | 49 |
| Silicium | 28 |
| Silicon Valley | 173 |
| SLSI | 28 |
| Sol 20. Processor Technology | 189 |
| somme de contrôle | 215 |
| « Sorcerer » de Exidy | 184 |
| Sord | 198 |
| Souris | 165 |
| Sous-programme | 82 |
| South West Technical Products | 202 |
| SQR | 82 |
| SSI | 28 |
| START | 240 |
| Stock | 98 |
| STOP | 240 |
| Structure de liste | 245 |
| Structuration d'un fichier | 344 |
| Subroutine | 82 |
| Sunnyvale | 173 |
| Super-imprimante | 157 |
| Sur mesure | 114 |
| Symboles | 232 |
| Synchrone | 240 |
| Système | 38 |
| Système binaire | 235 |
| Système d'exploitation | 133 |
| Systèmes de gestion | 205 |
| Système personnel | 127 |
| Système microordinateur | 18 |
| S 50 | 137 |
| S 100 | 137 |

T


TAB 85
Table ASCII 67
Tableaux 83
Tablette 165
Tabulation 84
Technologie 28
Télétype 55,150,240
Téléviseur 142
Télévision 144
Temps d'accès 49
Temps partagé 210
Terminal 18,147
Terminal à écran 18
Terminal «intelligent» 147
Paul Terrell 222
Test 28,80
Tests de performances 124
Tore 53
Touches 141
Traitements 98
Traitement de texte 100
Transistor 28
Trésorerie 98
TRS80 Radio-Shack 183
Tube cathodique 55
Tubes électroniques 171
TVA 97
Types de systèmes 130


U


UAL 31,50
UART 57
UC 47
Unibus 198
L'unité arithmétique 35
L'unité centrale 28,29,50,35,47
Unité de commande 31,131
L'unité de contrôle 50
Unité de visualisation 18,21
UNIVAC 172

V

Vallée du Silicium 173
Variables 77
Vector 1 de Vector Graphic 195
Verrou 233
VHF 142
Videobrain 176
Vitesse d'exécution 88
Vitesse du microprocesseur 123
Vitesses de transmission 241
Visualisation 21
Von Neumann 172

W

Wafers 32
Jim Warren 223
West Cost Computer Faire 223
Western Digital 200
Winchester disk 159

X

Xitan de Technical Design Labs 195
XPL 90

Z

Zilog 173
Z 80 137

6502 137
6800 137
8080 137